

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.043

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ” _____ 2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: Система рекомендації навчальних матеріалів для персоналізації навчального шляху

Виконав: студент другого курсу, групи ІТ-74мп
(шифр групи)

_____ Щербак Данило Вікторович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник ст.в., к.т.н., Сирота О.П.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

_____ (науковий ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

Рецензент доц., ктн., Ліщук К.І.

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2018 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики
Рівень вищої освіти – другий (магістерський)
Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
І.Р. Пархомей
(підпис)

«__» _____ 2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Щербак Данило Вікторович
(прізвище, ім'я, по батькові)

1. Тема дисертації «Система рекомендації навчальних матеріалів для персоналізації навчального шляху»,
науковий керівник дисертації ст. в., к.т.н. Сирота О. П., _____

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 2018 р. № _____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – рекомендаційні системи

4. Предмет дослідження – методи рекомендації матеріалів

5. Перелік завдань, які потрібно розробити – проаналізувати методи рекомендації матеріалів; розробка алгоритму та ПП;

6. Орієнтовний перелік ілюстративного матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – дві публікації

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	13.09.2018 р.	
2	Постановка задачі	15.09.2018 р.	
3	Аналіз інформаційного забезпечення	20.09.2018 р.	
5	Аналіз алгоритмічного забезпечення	25.09.2018 р.	
6	Розробка алгоритмічного забезпечення	15.10.2018 р.	
7	Розробка програмного забезпечення	01.11.2018 р.	
8	Маркетинговий аналіз стартап-проекту	10.11.2018 р.	
9	Висновки	15.11.2018 р.	

Студент

_____ (підпис)

Щербак Д.В.

(ініціали, прізвище)

Науковий керівник дисертації

_____ (підпис)

Сирота О.П.

(ініціали, прізвище)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з 4 розділів, містить 12 рисунків, 25 таблиць, 6 додатків, 15 джерел.

Дипломний проект присвячений аналізу рекомендаційних систем та розробці рекомендаційної системи навчальних матеріалів для навчальних закладів. Цілі розробки – зменшити інформаційне перевантаження студентів, рекомендуючи відповідний навчальний матеріал, що допоможе досягти їхніх цілей.

В рамках дослідження використовуються рекомендаційні методи та запропонований метод рекомендації навчальних матеріалів студентам є науковою новизною дослідження, що отримали відображення у розробленій системі. Автоматичний збір даних системою, їх аналіз та формування рекомендацій, як для студентів, стосовно навчальних матеріалів. Викладачам, рекомендуючи додати навчальні матеріали з тих тем, що цікаві студентам.

Ключові слова: рекомендаційний метод, графи переходів, веб додаток, фільтрація вмісту, колаборативна фільтрація, гібридна фільтрація.

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of 4 chapters, containing 12 figures, 25 tables, 6 applications, 15 sources.

The diploma project is devoted to the analysis of recommendation systems and the development of a reference system for teaching materials for educational institutions. The development goals will reduce student information overload by recommending appropriate training material to help them achieve their goals.

The research uses recommendation methods and the proposed method of recommending study materials to students is a scientific novelty of research, that was reflected in the developed system. Automatic data collection by the system, their analysis and the recommendation of study materials for students. Teachers, recommending adding educational materials on those topics that are of interest to students.

Keywords: recommendation method, conversion graphs, web application, content filtering, collaborative filtering, hybrid filtering.

Пояснювальна записка
до магістерської дисертації

на тему: *Система рекомендації навчальних матеріалів для
персоналізації навчального шляху*

Київ – 2018 року

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ....	12
1.1 Об'єкт та предмет дослідження.....	12
1.2 Огляд існуючих рішень, персоналізовані рекомендації.....	15
1.2.1 LinkedIn.....	15
1.2.2 Pluralsight.....	20
1.3 Постановка задачі.....	22
Висновки до розділу.....	24
РОЗДІЛ 2. АНАЛІЗ РЕКОМЕНДАЦІЙНИХ МЕТОДІВ ТА АЛГОРИТМ РОБОТИ.....	25
2.1 Рекомендаційні системи.....	25
2.1.1 Фільтрація контенту.....	25
2.1.2 Колаборативна фільтрація.....	26
2.1.3 Гібридна система фільтрації.....	27
2.2 Вибір рекомендаційного методу.....	27
2.2.1 Використання колаборативної фільтрації.....	27
Висновок до розділу.....	29
РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	30
3.1 Вимоги до системи.....	30
3.2 Архітектура.....	31
3.2.1 Збір даних.....	33
3.2.1.1 Явний збір даних.....	33

3.2.1.2 Неявний збір інформації.....	34
3.2.2 Засіб збереження даних.....	34
3.2.3 Мова програмування.....	38
3.2.4 Опис клієнтської частини.....	41
3.3 Алгоритм рекомендації матеріалів.....	44
3.3.1 Графи переходів.....	44
3.4 Реалізація системи.....	46
3.4.1 Обробники (способи рекомендації).....	47
3.4.2 Оцінка реакції користувача.....	50
3.4.3 Формування видачі.....	50
3.4.4 Лінійна регресія.....	52
3.5 Аналіз результатів.....	55
3.5.1 Прості рекомендації.....	57
3.5.2 Контекстні рекомендації.....	58
Висновки до розділу.....	59
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ.....	61
4.1 Опис ідеї проекту.....	61
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	63
4.4 Розроблення ринкової стратегії проекту.....	72
4.5 Розроблення маркетингової програми стартап-проекту.....	74
Висновки по розділу.....	77
ВИСНОВКИ.....	78
ПЕРЕЛІК ПОСИЛАНЬ.....	80
ДОДАТКИ.....	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface;

СУБД – Система управління базами даних;

HTML – HyperText Markup Language;

JS – JavaScript;

БД – база даних;

ПЗ – Програмне забезпечення;

SQL – Structured Query Language;

UML – Unified Modeling Language;

IDF – Inverse Document Frequency;

ВСТУП

В інтернеті постійно збільшується обсяг інформації і студентам все складніше знайти інформацію з відповідних навчальних матеріалів, що задовольнить їхні потреби. Ефективна особиста система рекомендації навчальних матеріалів не тільки зменшить інформаційне перевантаження, рекомендуючи відповідний навчальний матеріал для своїх студентів, а також надасть їм необхідну інформацію в необхідний час та запропонує необхідний шлях.

Кожен студент має різні вподобання й постійно зайнятий пошуком інформації у напрямку, що його цікавить. Для розвитку йому потрібно закріплювати існуючі та здобувати нові знання вивчаючи нові технології та методи розробки. Університети не можуть дати всі необхідні знання, по всіх предметних областях й для розвитку своєї кар'єри студенту потрібно отримувати нові навички, які в університеті не викладають, тому студенти витрачають багато часу на пошук цих знань.

В статті розглядається система, у якій студент буде обирати цікаву йому предметну область й система буде радити йому напрямки для розвитку у програмній області, яку він обрав. Студенти зможуть оцінювати свої поточні та

запропоновані шляхи розвитку й ця оцінка буде використовуватися для формування рекомендацій для інших студентів.

Існує багато рекомендаційних алгоритмів, що мають свої переваги та недоліки й підходять до відповідних задач.

У даній роботі буде представлено опис і аналіз існуючих систем рекомендацій і виявлення завдань для розробки та оцінка наборів даних. Потім розглянемо підхід для розробки рекомендаційної системи, метою якої є підтримка студента через систему управління навчанням, щоб знайти відповідний матеріал для покращення знань студента.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт та предмет дослідження

Велика кількість нині існуючих веб-сайтів використовують рекомендаційні системи для своїх користувачів. Вони генерують їм різні пропозиції, наприклад, супутні товари або рекомендують людей, зареєстрованих на цьому сайті, зі схожими інтересами. Такі рекомендаційні механізми обробляють величезні обсяги інформації для позначення потенційних переваг користувачів.

Рекомендаційні системи - це програми, які намагаються визначити що хочуть знайти користувачі, що може їх зацікавити, і рекомендують їм це. Ці механізми вдосконалили взаємодію між користувачем і сайтом. Замість статичної інформації вони надають динамічно змінюється: рекомендації генеруються окремо для кожного користувача, ґрунтуючись на його попередньої активності на даному веб-ресурсі, до того ж може враховуватися інформація, що виходить від інших відвідувачів.

У більшості випадків розвиток рекомендаційних систем полягає в поліпшенні алгоритмів рекомендації. Мета цього прогресу - давати відвідувачам сайтів найбільш точні рекомендації, що задовольняють їх запити в певний момент часу. Для досягнення цього математичні алгоритми, що лежать в основі рекомендаційних сервісів, повинні постійно вдосконалюватися. Тут вступає в

силу інтелектуальний аналіз даних Data mining. Схематично це можна описати так: сервіс веб-сайту надає користувачеві набір рекомендацій, далі розробник отримує від нього зворотний зв'язок, аналізує її на предмет відповідності даних, раніше рекомендацій інтересам відвідувача, перенавчати математичну модель, потім знову пропонує рекомендації і так далі по колу. Математичні алгоритми рекомендаційних систем розвиваєть по наступних аспектах:

1) кількість рекомендованого контенту - його повинно бути достатня кількість, щоб було що і з чого радити;

2) кількість інформації про рекомендований контенті (наприклад, хто автор книги, хто перекладач, хто ілюстратор, скільки символів в ній) - чим її більше, тим точніше ми можемо вибрати потрібний;

3) обсяг інформації про користувачів (стать, ім'я, вік, країна проживання) - знову ж таки, чим її більше, тим конкретніше будуть підібрані рекомендації;

4) зручність для користувача частини програми, тобто інтерфейсу - чим комфортніше для відвідувача, тим більше інформації ми можемо отримати про його реакції на наші поради.

Методи збору інформації, що надаються Інтернетом, значно спростили використання громадської думки за допомогою колаборативної фільтрації. Але, з іншого боку, великий обсяг інформації ускладнює втілення цієї можливості. Наприклад, поведінка одних людей досить ясно піддається моделюванню, в той час як інші поводяться абсолютно непередбачувано. І саме другі впливають на зміщення результатів рекомендаційної системи і зниження її ефективності. Інший

приклад: користувачі можуть використовувати рекомендаційну систему для навмисного зростання популярності одного продукту щодо іншого. Вони можуть залишати не просто позитивні відгуки про вподобаному товар, але і писати негативні про конкурентів.

Інша проблема, притаманна великим рекомендаційним системам, - масштабованість. Традиційні алгоритми непогано справляються з невеликою кількістю даних. Але чим більше інформації має оброблятися, тим важче отримати точні результати. Рекомендаційні системи, що працюють на основі неявних призначених для користувача оцінок, тягнуть за собою проблему дотримання конфіденційності користувачів, захисту персональних даних.

Ще однією проблемою є проблема холодного старту. Вона ділиться на два види: холодний старт по відношенню до користувача (що показувати тільки що зареєструвався людині?) і холодний старт по відношенню до сайтів (кому можна порекомендувати тільки створені сайти?).

Область застосування систем рекомендацій досить вузька, але важлива в сучасному інтернеті:

- Рекомендації товарів в інтернет-магазинах. Мабуть, цей варіант застосування на сьогодні найбільш актуальним; системи рекомендацій інтернет-магазинів пропонують користувачеві придбати потрібний, на думку алгоритму, товар, ґрунтуючись на різноманітних факторах;
- Рекомендації фільмів / музики / відео. Сервіси на зразок Netflix, YouTube або LastFM рекомендують зареєстрованим відвідувачам той чи інший

контент в залежності від їх попередніх переглядів або прослуховувань і / або рейтингів того чи іншого відео, фільму або виконавця;

- Рекомендації новин. Система рекомендує «схожі» на раніше прочитані матеріали новини, ґрунтуючись або на схожості ключових слів, або, знову ж таки, на виборі користувачів зі схожими інтересами.

1.2 Огляд існуючих рішень, персоналізовані рекомендації

1.2.1 LinkedIn

LinkedIn - це соціальна мережа для бізнес-спільноти. Заснована у 2003 році, онлайн-сайт - це місце для професіоналів, які можуть спілкуватися з минулими та нинішніми колегами, збільшувати кількість ділових зв'язків, мережу у своїй галузі, обговорювати бізнес-ідеї, шукати роботу та шукати нових робочих місць. Користувачі LinkedIn створюють професійні профілі, подібні до резюме, які дозволяють іншим учасникам сайту більше дізнатися про свій бізнес, спеціалісти та групи чи організації, до яких вони належать. Коли користувач створює свій профіль, він може додати інших користувачів до своєї мережі.

Нещодавно LinkedIn запустив LinkedIn Learning. Раніше їхні рекомендації базувалися на загальних та рандомізованих підходах, проте вони вирішили представити персоналізовані рекомендації, й саме вони допомогли їм покращити зацікавленість до свого ресурсу на 58%.

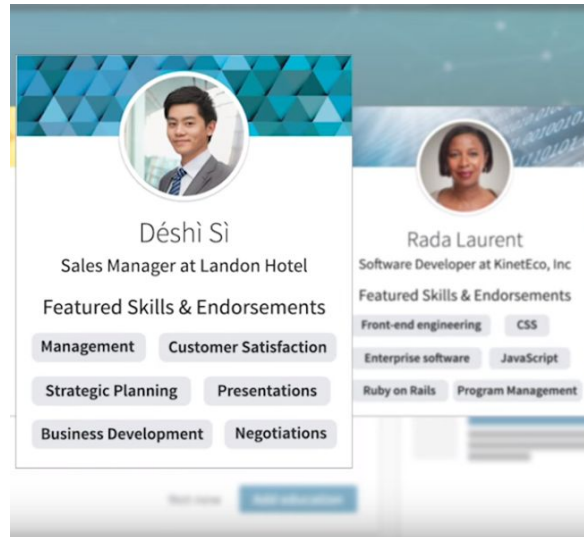


Рисунок 1.1 – Профіль користувача зі списком навичок та інтересів

При створенні нового система LinkedIn нічого про нього не знає й через це не може нічого порекомендувати, отож щоб уникнути цю проблему, після створення користувача йому пропонується відповісти на ряд питань, спрямованих на визначення його інтересів, а у випадку відмови, система буде використовувати дані фреймворків з якими працював користувач, зазначеними у його профілі.

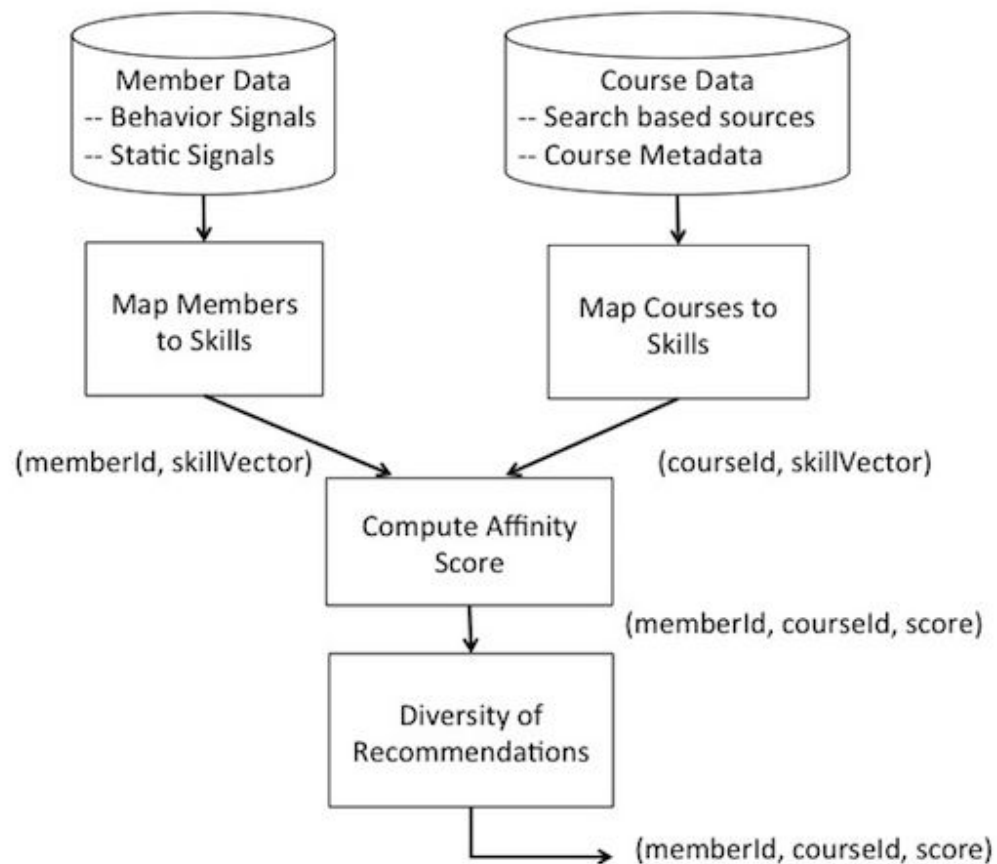


Рисунок 1.2 – Загальна архітектура рекомендаційної системи в мережі LinkedIn

Вони використовують два джерела даних: БД курсів та інформацію про користувачів. Їхня задача полягає у використанні навичок в якості властивостей для представлення користувачів та курсів. Курси можуть бути представлені векторами навичок, які можуть бути здобуті проходячи цей курс. Користувачі також можуть бути представлені в цьому просторі навичок, використовуючи різні джерела інформації.

Because you're interested in Coaching

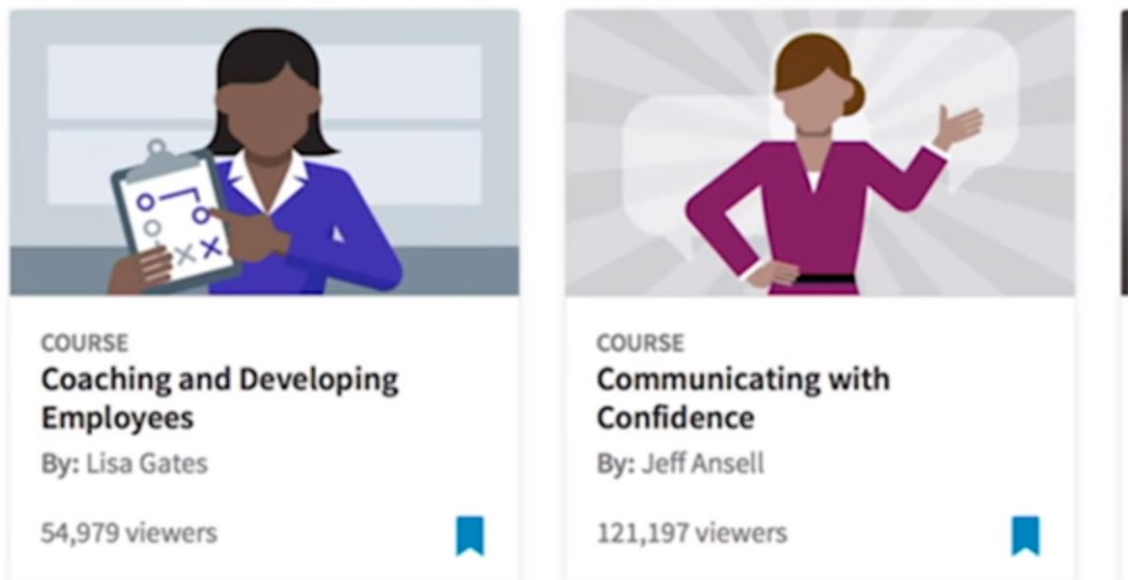


Рисунок 1.3 – Рекомендації у мережі LinkedIn, які були згенеровані використовуючи інтереси

В той час, як в профілі користувача явно задані навички, існують й інші джерела, що можуть використовуватися для виявлення не явних навичок, наприклад, може бути використані професійні зв'язки або активність користувачів у мережі. Як тільки користувачі та курси утворили простір навичок, система оцінює спорідненість між учасниками та параметрами курсів. Результуючі рекомендації для користувачів зберігаються в онлайнний та офлайнних сховищах.

Навички мають різну популярність. Для того щоб переконатися, що навичка, на якій базується рекомендований курс бере до уваги унікальні навички користувача, система визначає “зворотну частоту документу” (IDF) для кожної навички.



The soft skills companies need most – and how to learn them

First, let's start with the skills all professionals should learn, regardless of what they do. These are "soft" skills, although in practice they are anything but: 57 percent of leaders say soft skills are more important than hard skills.

1. Leadership

Recommended Courses: [Body Language for Leaders](#), [Strategic Thinking](#), [Leading Without Formal Authority](#)

2. Communication

Recommended Courses: [Communicating with Confidence](#), [Influencing Others](#), [Giving and Receiving Feedback](#)

3. Collaboration

Recommended Courses: [Effective Listening](#), [Building Business Relationships](#), [Finding Your Introvert/Extrovert Balance in the Workplace](#)

Рисунок 1.4 – Згенерований список популярних навичок станом на 2018 рік у мережі LinkedIn

Для підрахунку цієї частоти, користувацький профіль розглядається як вибірка слів, в цьому документі. Потім показник IDF розраховується як $IDF(s) = \log(1 + M/N(s))$, де M це загальна кількість користувачів, а $N(s)$ - це кількість користувачів, що мають навичку s у своєму профілі. Навички, що зустрічаються дуже рідко ($N(s) < 100$) не розглядаються. Вектору навичок, що представляє певного користувача, надається вага, яка розраховується за допомогою IDF, перед тим як використовуватися у кінцевому продукті, забезпечуючи механізм

обчислення зваженої подібності. Таким чином, користувач отримає рекомендовані курси на основі власних унікальних навичок.

Перед тим як представити рекомендації користувачу вони проходять додаткову обробку, що використовується для диверсифікації запропонованих курсів, щоб унеможливити присутність декількох схожих курсів, що розташовуються у верхній позиції списку. В якості прикладу, скажімо, що є список курсів, їх чотири, які мають набір навичок, позначених тегами (можуть бути представленні оновлені версії старшого курсу або той самий на іншому рівні складності). В усіх випадках показник схожості всіх чотирьох курсів буде однаковим для учасника, котрий має декілька або всі ті навички, зазначені в його профілі. У випадку, коли найвищий показник співвідношення відповідає цим курсам, то чотири основні рекомендації для користувача будуть чотирма курсами. Проблема включення різноманітності пропозицій являється класичною. В результаті використовують круговий алгоритм, щоб рекомендації були відмінними. Скажімо, у чотирьох курсах однакові навички й вони збігаються з поточними навичками користувача; в цьому випадку об'єднаємо ці курси в одну вибірку й додамо ознаку кожному курсу - скажімо, дату створення. Через це, отримавши навички, для котрих необхідна рекомендація з курсу, з вибірки береться останній курс. Після зроблення цієї вибірки, маємо маємо одну рекомендацію, створюється наступна вибірка з курсів, що має інший перелік навичок. Коли система дійшла до останньої вибірки, й взяла з неї останній курс, повертаємося до першої вибірки та беремо з неї наступний курс. Таким чином система уникає подібних курсів, рекомендованих по порядку.

1.2.2 Pluralsight

Pluralsight використовує експертний підхід, який полягає у формуванні каналів експертом та підписанні команди на нього. Експерт має можливість редагувати канали, надавати на них посилання. Також система представляє аналітику по тому, як часто команда користується каналом, середні та індивідуальні показники прогресу.

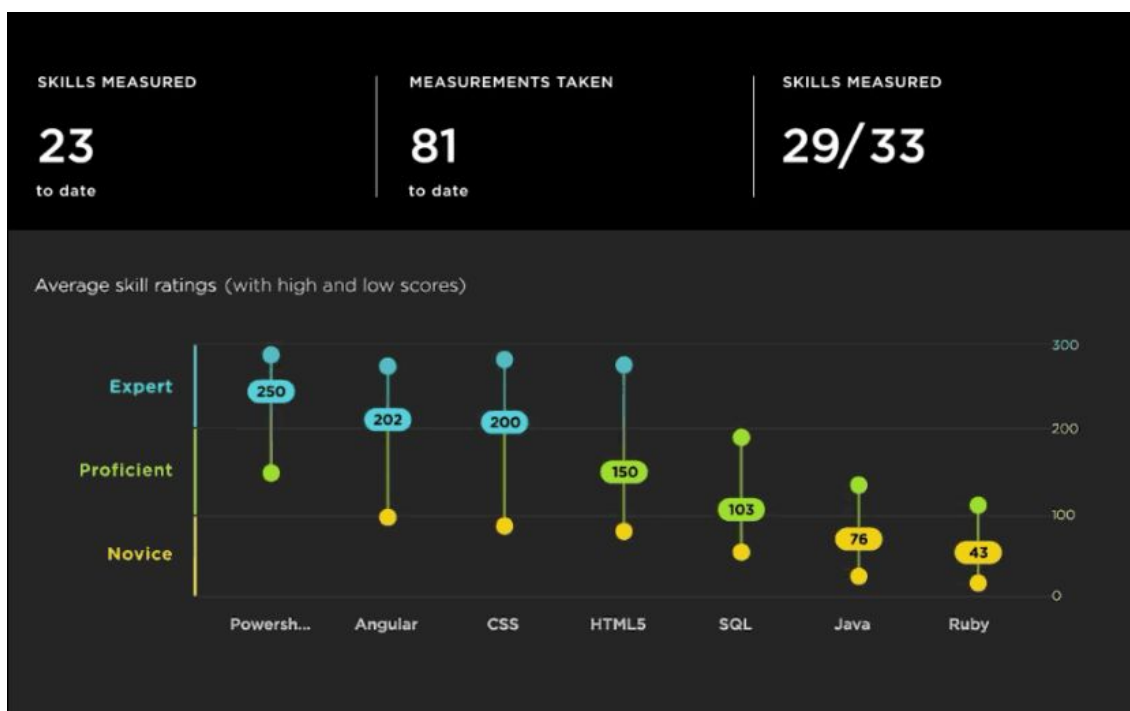


Рисунок 1.4 – Аналітика команди у мережі Pluralsight

Використовуючи таку аналітику менеджери та адміністратори зможуть отримати інформацію по сильних та слабких сторонах власної команди, визначити у яких напрямках їм буде необхідно розвивати свої навички. Командному менеджеру буде простіше планувати розробку й назначати на неї людей, враховуючи персональні навички кожного. Персональна аналітика навиків

базується на переоцінці, що являється ключовим фактором до поліпшення відслідкованості.

Yandex.Music перед тим як запропонувати музику, створює представлення о смаках користувача. Найпростіший спосіб зробити це - подивитися, які треки на «Яндекс.Музиці» він вже послухав. Це найважливіша інформація для рекомендаційної системи; з історії прослуховувань можна встановити, яких виконавців і які жанри людина вважає за краще. Однак щоб скласти більш повну картину, непогано ще розуміти, що йому подобається більше, а що - менше.

Для цього ми використовуємо додаткові дані. Одне з джерел таких даних - оцінки «Подобається» і «Не подобається», які ставлять користувачі.

Як правило, люди оцінюють музику, яка викликала у них сильний емоційний відгук - неважливо, позитивний або негативний. Тому оцінки досить точно відображають уподобання людини. Але одних оцінок недостатньо: по-перше, люди ставлять їх далеко не завжди, а по-друге, у шкалі не вистачає півтонів - є тільки або «добре» («Подобається»), або «погано» («Не подобається»).

Тому, крім оцінок і прослуховувань, ми звертаємо увагу і на інші дії користувача: пропуски треків і додавання треків в плейлисти.

Всі дії ми поділяємо на позитивні і негативні. Позитивні - прослуховування, оцінка «Подобається», додавання в плейлист - говорять про те, що музика подобається користувачеві, а негативні - пропуск і оцінка «Не подобається» - навпаки. Тому кожній дії ми присвоюємо вагу: у оцінки «Подобається» він максимальний, а у пропуску - мінімальний.

Алгоритм аналізує профіль користувача (тобто дані про його музичні уподобання) і передбачає, які треки і виконавці можуть йому сподобатися. Крім того, алгоритм вміє навчатися в режимі реального часу. Кожен раз, коли ви робите нову дію - слухаєте трек або додаєте його в плейлист, - профіль оновлюється, і прогноз будується заново. Це дозволяє швидко підлаштовуватися під смаки і пропонувати музику, яка відповідає сьогодишньому настрою.

Роблячи прогноз, алгоритм також враховує інформацію про те, як пов'язані один з одним об'єкти з каталогу «Яндекс.Музика»: треки, альбоми, виконавці, жанри. Завдяки цим даним можна порадити людині нових виконавців у його улюбленому жанрі. Крім того, система порівнює профілі всіх користувачів «Яндекс.Музика». Це робиться для того, щоб виявити людей зі схожими музичними вподобаннями: то, що подобається одному, може сподобатися і іншому.

1.3 Постановка задачі

Метою даної роботи є створення веб застосунку, який облегчить процес навчання для студентів. Кожен студент має різні вподобання, й постійно зайнятий пошуком інформації у напрямку, що їх цікавить. Для розвитку йому потрібно закріплювати існуючі та здобувати нові знання, вивчаючи нові технології та методи розробки. Університети не можуть дати всі необхідні знання, по всіх предметних областях й для розвитку своєї кар'єри студенту потрібно отримувати нові навички, які в університеті не викладають, тому студенти витрачають багато часу на пошук цих знань.

Першим етапом задачі є вибір методу рекомендацій, який краще за всіх підходить для досягнення мети та вирішення задачі, бо велика кількість нині існуючих веб-сайтів використовують різні рекомендаційні методи, для рекомендації контенту своїм користувачам. Існує багато рекомендаційних методів й кожен із них підходить для вирішення певних задач, тому необхідно визначити, який саме метод краще всього підійде для вирішення поставленої задачі.

Другий етап полягає у тому, щоб визначити алгоритм рекомендацій навчальних матеріалів у системі та визначити у якому вигляді будуть представлені навчальні матеріали й те, як користувачі взаємодіють з ним, тобто, яка буде підходити для вирішення задачі, та визначити, як вона буде використовуватися для генерування рекомендацій.

Третій етап задачі полягає у створенні системи, яка буде пропонувати студенту нові курси, спираючись на те, яку мету він хоче досягнути та враховуючи його рівень підготовки та знання. Крім цього рекомендації системи повинні бути адаптивними, які допоможуть студенту вивчати матеріал та гнучко підлаштовуватись під нього, й рекомендувати саме той контент, який потрібен йому даний момент часу. Система буде корисна користувачам своїми персоналізованими рекомендаціями, які допоможуть студенту вивчити певну тему й досягти свою ціль. В систему будуть вноситися дані користувачем, які засвідчують його цілі. Враховуючи цілі користувача, система буде рекомендувати навчальні матеріали по проходженню яких студент матиме змогу оцінювати та доповнювати їх й ця оцінка буде використовуватися при формуванні рекомендацій для інших студентів.

Після генерації рекомендацій, система може запропонувати декілька варіантів, й при виборі його користувачем, враховувати цей вибір для наступної генерації.

Висновки до розділу

У першому розділі було визначено, що предметом дослідження являються рекомендаційні системи. За допомогою проведення аналізу існуючих рішень було виявлено основні сучасні рекомендаційні системи та проведений їхній аналіз.

Була сформована задача, яка буде проаналізована під час роботи з магістерською дисертацією. Було вказано призначення розробки рекомендаційної системи, цілі та задачі, які повинні бути досягнуті під час роботи з системою рекомендацій навчальних матеріалів.

РОЗДІЛ 2. АНАЛІЗ РЕКОМЕНДАЦІЙНИХ МЕТОДІВ ТА АЛГОРИТМ РОБОТИ

2.1 Рекомендаційні системи

Аналіз математичних моделей для персоналізованих рекомендацій. Рекомендаційні системи - це програмні засоби на основі машинного навчання та рекомендаційних алгоритмів, які надають пропозиції, щодо потенційно корисної інформації для студента.

Існують основні стратегії створення рекомендаційних систем:

- Фільтрація вмісту
- Колаборативна фільтрація
- Гібридна система фільтрації

2.1.1 Фільтрація контенту

Рекомендаційні системи, що базуються на фільтрації контенту передбачають цікавий користувачу елемент, опис якого схожий на модель властивості користувача. Ця система працює з користувачами і профілями об'єктів з якими користувач взаємодівав у минулому. Профілі взаємодії представлені векторами, які отримують характеризуючі атрибути об'єктів або користувачів. Прототип такої рекомендаційної системи був запущений у 2014, але розробники зіштовхнулися з проблемою, що рекомендації були згенеровані для всіх студентів одноманітні і їм бракувало персоналізації.

2.1.2 Колаборативна фільтрація

Техніки колаборативної фільтрації найчастіше використовуються у соціальних мережах і медіа ресурсах, що пропонують цікаві спільні ресурси. Метою даної системи є оцінка індивідуальних преференцій і пропозиція посилань на інші ресурси або інші системи, продукти й ресурси, які вірогідно будуть цікаві користувачу. Основне допущення полягає в наступному: ті, хто однаково оцінювали будь-які курси в минулому, схильні давати схожі оцінки інших предметів і в майбутньому. Наприклад, за допомогою колаборативної фільтрації музичний додаток здатний прогнозувати, яка музика сподобається користувачу, маючи неповний список його переваг (симпатій і антипатій). Прогнози складаються індивідуально для кожного користувача, хоча використовувана інформація зібрана від багатьох учасників. Тим самим колаборативна фільтрація відрізняється від більш простого підходу тим, що дає усереднену оцінку для кожного об'єкта інтересу, наприклад, що базується на кількості поданих за нього голосів. Обмеження даної системи в тому, що вона об'єднує в групи користувачів, у котрих схожий стиль навчання й навички та пропонує їм одне й те саме, не враховуючи особливості кожного. У 2015 році була створена навчально колаборативна система фільтрації під назвою U Learn (<http://ulearn.education>). Ця система забезпечує пропозиціями навчальних матеріалів, викладення їх найкоротшим шляхом, спрощує переміщення по навчальному ресурсі. Проблемою даної системи є те, що в ній користувачі не можуть взаємодіяти з матеріалом, доповнюючи його, тим самим допомагаючи один одному краще зрозуміти його.

2.1.3 Гібридна система фільтрації

Гібридна система фільтрації включає комбінує наступні техніки рекомендації колаборативну та фільтрацію на основі контенту. Існує багато типів гібридних рекомендаційних систем, але основними є наступні типи:

- колаборативна з доданими можливостями взятими у техніки фільтрації контенту
- фільтрація контенту з використаними техніками колаборативної фільтрації.

2.2 Вибір рекомендаційного методу

У даній роботі буде обрана гібридна система з більш активним використанням фільтрації контенту і менш активною - колаборативною фільтрацією. Колаборативна фільтрація має обмеження, рекомендації, що вона створює, підходять до користувачів з однаковим стилем навчання. З іншої сторони метод фільтрація контенту підходить для більш персоналізованої рекомендацій користувачу курсів у додатку, вона буде визначати з якими об'єктами він взаємодіяв у минулому, аналізувати його профіль й на основі цього пропонувати матеріал.

2.2.1 Використання колаборативної фільтрації

Система використовує різні підходи для пошуку контенту, який необхідний студенту. Кожен спосіб буде виділений в окрему функцію, яка дістає список навчальних матеріалів використовуючи вбудовану в неї логіку визначення рекомендаційної ваги, в залежності від того, наскільки навчальні матеріали підходять користувачу. Для цього будемо використовувати граф переходів,

вершинами якого будуть представлені одиниці контенту, а вага дуг буде показувати кількість переходів між ними. Дії які користувач здійснив у додатку, зберігатимуться у БД. Кожна дія матиме час, та буде розділена на типи, відповідно до дій користувача, наприклад, обрав контент, завершив, лишив відгук та оцінку. Візуально граф переходів представлятиме ряд вершин - навчальні матеріали, між деякими з яких є дуги. Наявність дуги між контентом означає, що існує користувач, який переглядав цей контент один за одним.

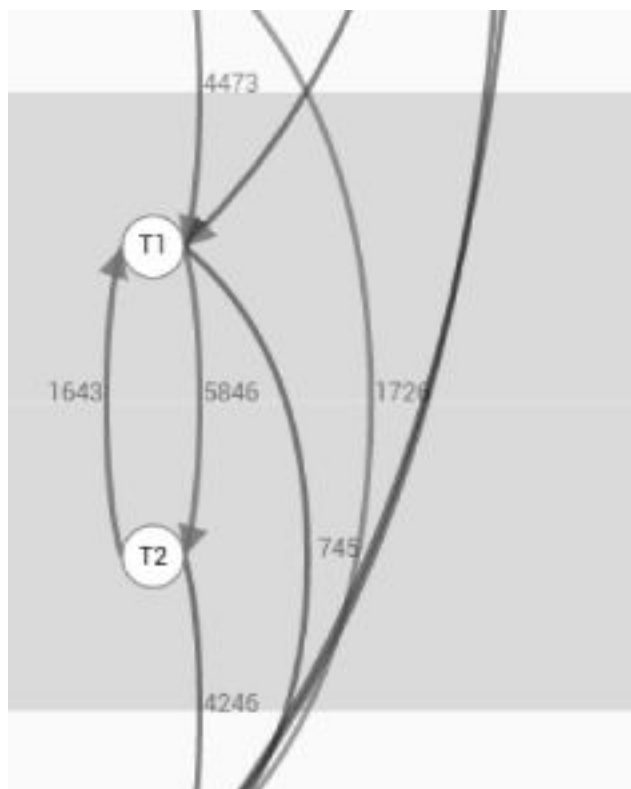


Рисунок 2.1 – Граф переходів для однієї теми

Процес створення графу складається з наступних кроків: спочатку для кожного користувача ми розглядаємо всі його дії по відношенню до навчальних матеріалів, сортуємо їх за часом, створюємо зв'язки, в нашому випадку переходи між контентом та підраховуємо для кожного переходу його вагу враховуючи як

користувач взаємодіяв з наступним контентом, він міг перейти й ознайомитися з описом, почати та не закінчити, почати й повністю опрацювати навчальний матеріал, залишити оцінку. Для отримання загального графу переходу, сумуємо частоти для всіх користувачів.

Висновок до розділу

У другому розділі даної роботи були проаналізовані рекомендаційні системи, обрана рекомендаційна система, що підходить для вирішення задачі

Додатково було розглянуто питання формування рекомендацій та описаний загальний метод їхнього формування.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вимоги до системи

Кожен студент має різні вподобання, й постійно зайнятий пошуком інформації у напрямку, що їх цікавить. Для розвитку йому потрібно закріплювати існуючі та здобувати нові знання, вивчаючи нові технології та методи розробки. Університети не можуть дати всі необхідні знання, по всіх предметних областях й для розвитку своєї кар'єри студенту потрібно отримувати нові навички, які в університеті не викладають, тому студенти витрачають багато часу на пошук цих знань.

Вимоги до системи та експеримент буде полягати у тому, щоб створити систему, яка на відміну від LinkedIn Learning та Pluralsight буде орієнтована на навчальні заклади. У системі модераторами будуть створені базові навички та теми, що входять у ці навички. Користувачі, що відповідають за завантаження курсів, у нашому випадку викладачі, матимуть змогу створити курс, використовуючи при цьому власний чи сторонній матеріал. Після створення курсу викладачі пропоставляють теми, які покриває їхній курс.

Користувачами нашої системи будуть студенти, вони матимуть змогу переглядати список курсів, проходити їх та отримувати навички, що будуть відображатися у їхньому профілі. Під час проходження курсу, студентам може бути не зрозумілий деякий матеріал, й вони будуть користуватися пошуковими системами, для знаходження відповідей на свої запитання. На відміну від LinkedIn

у нашій системі студенти зможуть додати знайдену відповідь до курсу. Всі ці доповнення будуть знаходитись у БД, й по проходженню подібної теми іншим студентом, система буде рекомендувати йому додаткову інформацію, що завантажив інший студент. По проходженню курсу, студент може оцінювати, як сам курс так й рекомендований додатковий матеріал інших студентів. Система буде збирати не лише оцінки користувачів, а й користувацькі дії, такі як перегляд курсів, перегляд доповнень, тощо. Вона буде рекомендувати студенту теми, з яких йому було би цікаво отримати навички, або він буде обирати теми власноруч, після чого система запропонує курси, які допоможуть отримати необхідні знання. А викладачам система запропонує створити нові курси, на основі популярних тем для студентів. Дана система буде відкритою і це означає, що додавати контент зможуть як викладачі так і студенти. Викладачі будуть створювати курси та додавати різні статті, а студенти в свою чергу матимуть можливість доповнювати курси, роблячи матеріал зрозумілішим.

3.2 Архітектура

Для розробки веб додатку буде використовуватися чиста архітектура або багат шарову. Додатки, що використовують методи інверсії залежностей і проблемно-орієнтованого проектування, мають схожу архітектуру. Протягом багатьох років вона носила найрізноманітніші назви. Спочатку це була шестигранна архітектура, на зміну якій прийшла архітектура портів і адаптерів. На сучасному етапі вона називається багат шаровою або чистою архітектурою.

В рамках чистої архітектури центральним елементом програми є його бізнес-логіка і модель. У цьому випадку бізнес-логіка не залежить від доступу до даних або іншим інфраструктур, тобто стандартна залежність інвертується:

інфраструктура і деталі реалізації залежать від ядра програми. Це досягається шляхом визначення абстракцій або інтерфейсів в ядрі додатки, які реалізуються типами, визначеними в шарі інфраструктури. Таку архітектуру зазвичай малюють у вигляді серії кіл із загальним центром, яка зовні нагадує зріз цибулини. На рис. 3.1 показаний приклад такого стилю уявлення архітектури.

Clean Architecture Layers (Onion view)

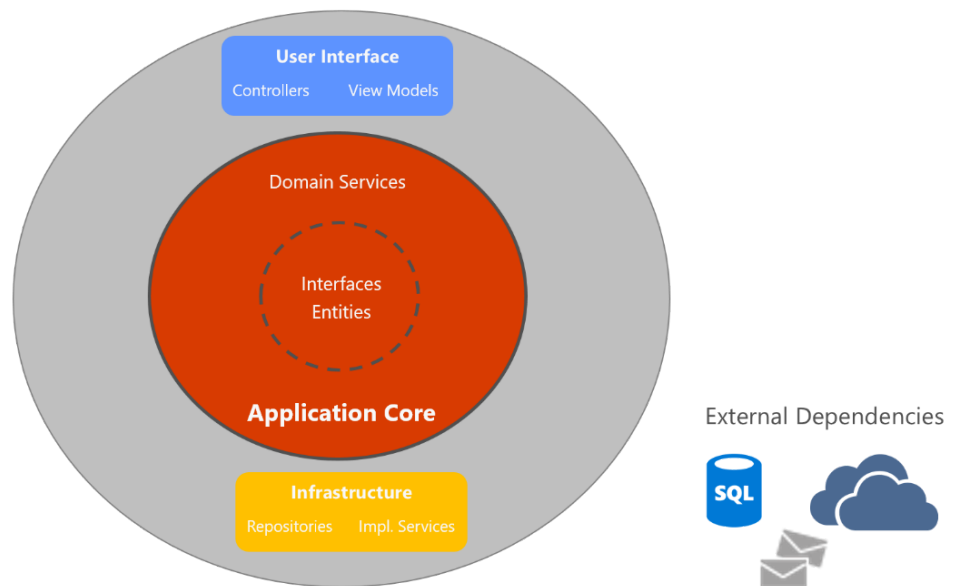


Рисунок 3.1 – Чиста архітектура

На цій схемі залежності спрямовані з самої внутрішньої окружності. Ядро програми називається так тому, що знаходиться в самому центрі цієї схеми. Як видно на схемі, ядро програми не має залежностей від інших верств додатки. Суті і інтерфейси додатків ви можете знайти в самому центрі. Відразу після них, але все ще в межах ядра додатка, розташовані доменні служби, які зазвичай реалізують інтерфейси, визначені у внутрішньому колу. За межами ядра додатки

розташовуються шари призначеного для користувача інтерфейсу і інфраструктури, які залежать від ядра додатка, але не друг від друга (обов'язково).

3.2.1 Збір даних

У системі буде представлено декілька ролей, адміністратор, студент, викладач, який виконує роль модератора та контент менеджера. Адміністратори відповідають за надання прав користувачам. Викладачі мають змогу створювати навчальний контент (курси, статті, мануали, тощо) та переглядати по ним статистику. Студенти мають змогу проходити курси, оцінювати їх, доповнювати інформацію по даній темі.

Будь який алгоритм рекомендацій передбачає первинний збір даних. В цьому і криється лівова частка «підводних каменів» роботи систем рекомендації. Дані у систему будуть додаватися за наступними методиками: явно і неявно.

3.2.1.1 Явний збір даних

Як випливає з назви, в цьому випадку користувач добровільно надає системі необхідні для роботи дані. Для нашого ПП явний збір даних може складатися з внесення викладачами загальних тем та навчальних матеріалів, що відповідають одній чи кільком темам. Зі сторони студента це внесення у профіль тем, які він хоче опанувати та теми, що просто цікаві йому. Студент матиме можливість доповнити інформацію з будь якої теми. Також система буде отримувати оцінки контенту, які пройшов студент.

3.2.1.2 Неявний збір інформації.

Ця методика збору даних являє собою своєрідне «шпигунство» за користувачем - в нашому прикладі неявний збір інформації відбувається за допомогою спостереження за поведінкою користувача - включаючи його переходи між контентом й як він з ним взаємодіє.

3.2.2 Засіб збереження даних

Перед розробкою ПЗ необхідно визначити СУБД, яка задовольняє всі вимоги ТЗ. Проаналізувавши ТЗ, стало зрозуміло, що вимогам відповідає реляційна SQL БД - MySQL та документо-орієнтована NoSQL БД - MongoDB.

MongoDB - це крос-платформа, орієнтована на документи база даних, яка забезпечує високу продуктивність, високу доступність і легку масштабованість. MongoDB працює над концепцією колекціонування та документацією.

MySQL DB (реляційна бд) - це сукупність відносин, що містять всю інформацію, яка повинна зберігатися в БД. Однак користувачі можуть сприймати таку базу даних як сукупність таблиць. Варто відмітити, що кожна таблиця складається з однотипних строк і має унікальну назву; Строки мають фіксоване число полів і значень. Інакше кажучи, в кожній позиції таблиць на перетині рядків і стовпчиків завжди маєтись в точності одне значення або ніщо;

При використанні MongoDB БД, дані що знаходяться в колекції симетричні з даними, що ми зберігаємо у неї, а у MySQL БД, різниця між даними, що зберігаються й даними у БД, зростає в залежності від складності додатку. На рисунку 1.1 показаний приклад відображення даних у SQL та NoSQL БД, з ліва

показано як дані користувача відображаються у таблицях в MySQL, а з права показано як дані користувача відображаються у колекції в MongoDB.



Рисунок 2.1 – Відображення даних у SQL та NoSQL БД

Звичайно для того щоб зберегти дані користувача та номер телефону в MySQL потрібно лише 2 таблиці, але від складності сутності цих таблиць може бути більше. Порівняємо властивості цих БД.

У таблиці 1.1 наведено порівняння властивостей двох баз даних, MySQL та Mongo. Враховуючи те, що складні транзакції не потрібні для розробляемого застосунка, була обрана MongoDB в якості БД.

Таблиця 1.1 – Порівняння властивостей баз даних: MySQL та Mongo

	MySQL	MongoDB
Багаті моделі даних	Ні	Так
Динамічні колекції	Ні	Так
Типізовані дані	Так	Так
Оновлення полів	Так	Так
Легкість у розробці	Ні	Так
Складні транзакції	Так	Ні
Аудит	Так	Так
Автоматичне сегментування (розподілення БД на декілька серверів)	Ні	Так

MongoDB це документо-орієнтована система керування базами даних (СКБД) з відкритим сирцевим кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінга (розподіл набору даних по серверах на основі певного ключа), комбінуючи який реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера в кластер проводиться без зупинки роботи БД простим додаванням нових машин.

Основні можливості MongoDB:

- Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних)
- Досить гнучка мова для формування запитів
- Динамічні запити
- Повна підтримка індексів
- Профілювання запитів
- Швидкі оновлення «на місці»

- Ефективне зберігання двійкових даних великих обсягів, наприклад, фото та відео
- Журналювання операцій, що модифікують дані в БД
- Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг
- Може працювати відповідно до парадигми MapReduce

3.2.3 Мова програмування

Серверна частина застосунку написана на мові програмування JAVA. **Java** — об'єктно-орієнтована мова програмування, мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Мова програмування Java розроблялася як платформа для програмування різноманітних пристроїв. Проте стрімкий розвиток інтернет-технологій змусив творців java працювати і в цьому напрямку.

Сфера застосування JAVA являє собою наступну картину:

- Створення десктопів і аплетів. Даний продукт створюється на замовлення.
- Створення мобільних додатків. Окремий світ програмування у всесвіті JAVA.

- Різні серверні додатки, які в основному орієнтовані на роботу з мережею.

Для розробки web застосунку використовувалось розширення Spring Framework.

Spring Framework — це фреймворк з відкритим кодом та контейнера з підтримкою інверсії управління для платформи Java.

Основні особливості Spring Framework можуть бути використані будь-яким додатком Java, але є розширення для створення веб-додатків на платформі Java EE. Незважаючи на це, Spring Framework не нав'язує якоїсь конкретної моделі програмування, Spring Framework став популярним в спільноті Java як альтернатива, або навіть доповнення моделі Enterprise JavaBean (EJB). Spring Framework складається з декількох модулів, які надають широкий спектр послуг:

- Контейнер Інверсії управління (IOC Container): Конфігурація компонентів додатків і управління життєвим циклом об'єктів Java, здійснюється головним чином через Інверсію управління
- Аспектно-орієнтоване програмування (AOP): дозволяє реалізувати наскрізні процедури
- Доступ до даних (Data): робота з реляційною системою управління базами даних на платформі Java з використанням JDBC і об'єктно-реляційні відображення та інструментів з NoSQL баз даних
- Управління транзакціями: об'єднує кілька API, управління транзакціями та координує операції для Java-об'єктів

- Модель-Вигляд-Управління (Model-View-Controller): програмний каркас на основі HTTP сервлета, що забезпечує створення веб-додатків і веб-служб RESTful.
- Аутентифікація і авторизація: настроювані процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик за допомогою підпроєкту Spring Security (колишня система безпеки Ace для Spring).
- Віддалене управління: конфігураційний вплив і управління Java-об'єктами для місцевої (локальної) або віддаленої конфігурації через JMX
- Тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів

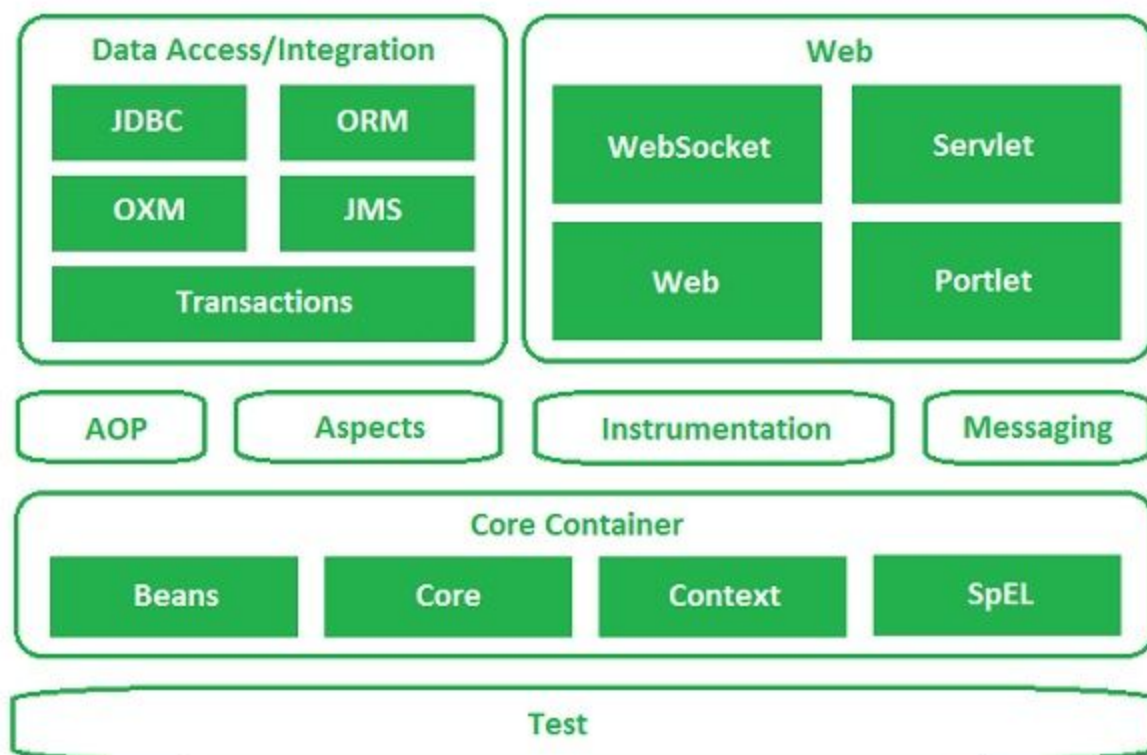


Рисунок 2.2 – Архітектура Spring Framework

Для генерування статичної API документації використовується Swagger.

Swagger — являє собою проект, що використовується для опису і документування RESTful APIs. Специфікація Swagger визначає перелік файлів, що необхідні для опису API та згенерувати його для клієнтів на різних мовах. Скористатися описом API можуть додаткові утиліти, наприклад утиліти для тестування.

Для перегляду стеку технологій та їхніх метрик використовується DropWizard Metrics.

DropWizard Metrics — у середовищі розробки надає інструментарій для вимірювання поведінки критичних компонентів. Надає вам видимість повного стека технологій для розповсюджених бібліотек, таких як Jetty, Logback, Log4j, Apache HTTPClient, Ehcache, JDBI, Jersey, Metrics.

Для створення HTML шаблонів використовується Thymeleaf.

Thymeleaf — сучасний серверний Java движок шаблонів для веб і автономних середовищ. Основною метою Thymeleaf є принести елегантні природні шаблони для робочого процесу розробки - HTML, який може бути коректно відображатися в браузерах, а також працювати в якості статичних прототипів, що дозволяє більш тісної співпраці в командах розвитку.

3.2.4 Опис клієнтської частини

Для розробки веб сторінок використовується JavaScript.

JavaScript — динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати

браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки, проте його також використовують багато небраузерних середовищ, наприклад node.js та Apache CouchDB. JavaScript є прототип-орієнтованою, динамічною скриптовою мовою з декількома парадигмами із підтримкою об'єктно-орієнтованого, імперативного й декларативного (функціональне програмування) стилів.

Використовуються наступні JavaScript фреймворки:

- AngularJS
- JQuery

AngularJS — це JavaScript-фреймворк з відкритим програмним кодом, який розробляє Google. Призначений для розробки односторінкових додатків, що складаються з одної HTML сторінки з CSS і JavaScript. Його мета — розширення браузерних застосунків на основі шаблону Модель-вид-контролер (MVC), а також спрощення їх тестування та розробки.

Фреймворк працює зі сторінкою HTML, що містить додаткові атрибути і пов'язує області вводу або виводу сторінки з моделлю, яка є звичайними змінними JavaScript. Значення цих змінних задаються вручну або отримуються зі статичних або динамічних JSON-даних.

AngularJS спроектований з переконанням, що декларативне програмування найкраще пасує для побудови інтерфейсів користувача та опису програмних компонентів, в той час як імперативне програмування пасує для опису бізнес-логіки. Фреймворк адаптує та розширює традиційний HTML, щоб забезпечити двосторонню прив'язку даних для динамічного контенту, що дозволяє автоматично синхронізувати модель та вид. У результаті AngularJS

зменшує роль DOM-маніпуляцій з метою підвищення продуктивності і спрощення тестування.

jQuery — популярна JavaScript-бібліотека з відкритим сирцевим кодом. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день. jQuery є вільним програмним забезпеченням під ліцензією MIT. Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

Для створення динамічних сторінок використовується Bootstrap.

Bootstrap — це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-застосунків, є найбільш популярним HTML, CSS, та JS фреймворком для зовнішнього вигляду та інтерактивності веб-сторінок. Bootstrap допомагає швидше та простіше розробляти зовнішній вигляд веб-сторінок. Він підходить для людей з будь-яким рівнем досвіду, для пристроїв будь-яких форматів, та проектів будь-якого розміру.

3.3 Алгоритм рекомендації матеріалів

В процесі навчання користувач залишає значний цифровий слід. Крім інформації про його прогрес зберігаються також відомості про те, коли він відвідував різні навчальні матеріали, які оцінки ставив, як з ним взаємодівав, чи завершував його та інформація про послідовність вибору матеріалів.

Найцікавіші для базової рекомендаційної системи такі відомості: теми, якими користувач цікавився, навчальні матеріали, на які він заходив, та які він пройшов, а також, якщо рекомендація контекстна, поточний навчальний матеріал.

Крім інформації про конкретного користувача, збережені дані можна розглядати як відомості про взаємодію користувачів з контентом. Одним із способів агрегованих і використовувати ці відомості є графи переходів, про яких детально розказано в наступному параграфі.

3.3.1 Графи переходів

Існує необхідність дізнатися, як користувачі вивчають контент, чим частіше цікавляться, які уроки пропускають, а на які повертаються для закріплення знань й у якому порядку їх проходять. Використовуючи збережені дані о діях користувачів, не важко дати відповідь на ці питання.

Для цього будемо використовувати граф переходів, вершинами якого будуть представлені одиниці контенту, а вага дуг буде показувати кількість переходів між ними. Дії які користувач здійснив у додатку, зберігатимуться у БД. Кожна дія матиме час, та буде розділена на типи, відповідно до дій користувача, наприклад, обрав контент, завершив, лишив відгук та оцінку. Візуально граф

переходів представлятиме ряд вершин - навчальні матеріали, між деякими з яких є дуги. Наявність дуги між контентом означає, що існує користувач, який переглядав цей контент один за одним.

Процес створення графу складається з наступних кроків: спочатку для кожного користувача ми розглядаємо всі його дії по відношенню до навчальних матеріалів, сортуємо їх за часом, створюємо зв'язки, в нашому випадку переходи між контентом та підраховуємо для кожного переходу його вагу враховуючи як користувач взаємодіяв з наступним контентом, він міг перейти й ознайомитися з описом, почати та не закінчити та почати й повністю опрацювати навчальний матеріал та залишити оцінку. Для отримання загального графу переходу, сумуємо частоти для всіх користувачів.

Використовуючи графи переходів, система може генерувати статистику по навчальним матеріалом, щоб викладачі могли ознайомитися з нею та проаналізувати навчальні матеріали, подивитися, в якому порядку користувачі дивляться матеріали, що їх більше цікавить, а до чого вони байдужі. Для простоти аналізу за темою обчислювалась середня частота переходу між навчальними матеріалами та ребра в підсумковому графі фарбувалися зеленим, якщо перехід по ним відбувався частіше очікуваного, або червоним, якщо рідше. Фрагмент графа переходів по одному з курсів представлений на рис. 3.2. З цього фрагменту видно, що крок з теорією T4 частіше звичайного пропускають і переходять до наступного за ним.

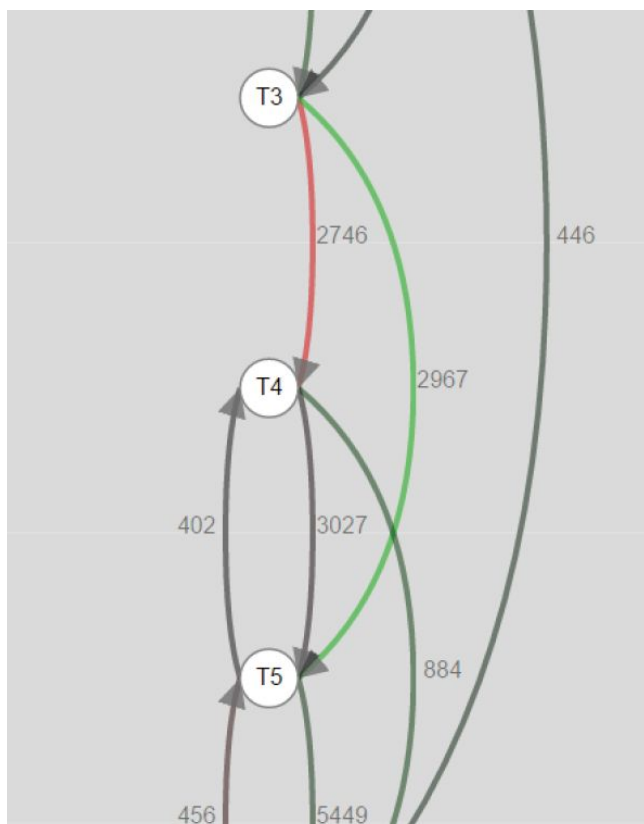


Рисунок 3.2 – Граф переходів для однієї теми з визначеною популярністю

3.4 Реалізація системи

В рамках даної роботи була створена рекомендаційна система, що пропонує користувачу навчальні матеріали на основі того, чим він цікавився раніше. Система використовує різні підходи для пошуку контенту, який необхідний студенту. Кожен спосіб буде виділений в окрему функцію, обробник, яка дістає список уроків використовуючи вбудовані в неї принципи, визначення ваги в залежності від того, наскільки вони підходять користувачу. Детальніше про ці обробники буде далі.

Рекомендаційна система може бути використана в різних ситуаціях. Основні випадки:

- рекомендація для конкретного студента. Користувач може побачити їх на головній сторінці додатку, також іноді він буде отримувати електронного листа.
- контекстна рекомендація, яка показує студенту напрямки які можуть його зацікавити, ця рекомендація буде відображатися користувачу коли той завершить курс.
- рекомендації викладачам. Рекомендація заключається у тому, щоб представити перелік популярних тем, які цікаві студентам й запропонувати додати навчальні матеріали.

3.4.1 Обробники (способи рекомендації)

У цьому параграфі будуть перераховані функції-Обробники, кожна з яких реалізує свій спосіб рекомендацій. Обробник приймає в якості параметрів користувача, а також тему, на якому той перебуває, якщо рекомендація контекстна. Обробник повертає для користувача j список навчальних матеріалів, в якому кожному курсу відповідає вага $weight_{ij} \in [0; 1]$, чим вона більша, тим краще рекомендація: $[(i; weight_{ij})]$. Нехай ми хочемо порекомендувати користувачеві j навчальний матеріал по конкретній темі, для якого частка пройдених користувачем курсів становить $tag_progress_j$. У разі, якщо за нього немає шляхів, всі його навчальні матеріали отримують вагу $tw_{ij} = tag_progress_j$. Якщо ж шлях є, то для кожного курсу ця вага ділиться на відстань до

найближчого попереднього в шляху навчального матеріалу, пройденого користувачем j : $tw_{ij} = tag_progress_j / dist_{ij}$. Таким чином реалізується ідея, що користувачеві може бути цікаво проходити матеріали, розташовані в дорозі недалеко від уже вивчених їм.

- Навчальні за цікавими користувачеві темами - обробник радить навчальні матеріали, що відповідають темам, які користувач уже вивчав. Для цього використовується описана вище рекомендація по тегам, вага курсу i для користувача j становитиме $weight_{ij} = tw_{ij}$.

- Незакінчені навчальні матеріали - радяться матеріали, які користувач починав і не закінчив, з вагою тим більшим, чим більшу частину матеріалу користувач пройшов: $weight_{ij} = course_progress_{ij}$, частка курсу i , пройденого користувачем j .

- Популярні навчальні матеріали - обробник не використовує інформацію про користувача, а радить просто найпопулярніші за останню тиждень курси на платформі. Вага прямо пропорційний популярності: $weight_{ij} = 1 / n_i$, де n_i - номер навчального матеріалу в списку популярних курсів.

- Шляхи по темам - використовуються шляхи, що містять навчальні матеріали, які користувач проходив, радяться навчальні матеріали слідом за пройденими, з вагою тим меншим, чим далі навчальний матеріал від уже пройдених: $weight_{ij} = 1 / dist_{ij}$, де $dist_{ij}$ - відстань від курсу i до найближчого попереднього в шляху навчального матеріалу, пройденого користувачем j .

- Граф переходів по темам - рекомендуються навчальні матеріали за темами, що цікавлять користувача. Ваги залежать від прогресу користувачів за темами, а

також від відносної частоти переходів між навчальними матеріалами. Нехай f_{req} – частота переходу від теми, курси якого користувач вивчив, до деякої теми t . Тоді для курсу i , позначений темою t , вага рекомендації для користувача j буде $weight_{ij} = freq \cdot tw_{ij}$.

- Навчальні матеріали по темам від схожих користувачів - на основі того, що цікавлять користувача, виявляються користувачі зі схожими темами, і рекомендуються навчальні матеріали за темою, які вони самі вивчають. Нехай у поточного користувача j знайшовся схожий на нього користувач k , причому міра схожості між ними - s (лежить між 0 і 1, чим вона більша, тим більше схожі користувачі). Тоді навчальний матеріал i , позначений темою, який вивчав k , будуть радитися з вагою $weight_{ij} = s \cdot tw_{ij}$. Цей обробник реалізує спосіб рекомендацій з допомогою колаборативних фільтрації.

- Граф переходів по навчальних матеріалах (тільки при контекстних рекомендаціях) - обробник радить навчальні матеріали, які йдуть за поточним в графі переходів, вага залежить від відносної частоти переходу між поточним навчальним матеріалом і відповідним. Якщо перехід з поточного навчального матеріалу в інший i відбувається з відносною частотою $freq_i$, то $weight_{ij} = freq_i$.

- Шляхи по навчальним матеріалом (тільки при контекстних рекомендаціях) - радяться навчальні матеріали, наступні за поточним в будь-яких шляхах, ваги обернено пропорційні відстані між навчальними матеріалами: $weight_{ij} = 1 / dist_{ij}$, де $dist_{ij}$ - відстань від навчального матеріалу i до поточного.

- Якщо всі попередні обробник в сукупності порадили менше курсів, ніж було потрібно, в рекомендації додаються випадкові курси, вага таких курсів дорівнює 0.

3.4.2 Оцінка реакції користувача

Щоб зрозуміти, наскільки успішна виявилася рекомендація, після її показу фіксується, чи перейшов користувач за посиланням, а також яку частину від навчального матеріалу користувач пройшов. У той же час у користувача є можливість дізнатися, чому навчальний матеріал був запропонований (варіанти відповідають Обробникам) і позначити рекомендацію, яка не цікава йому. Також ми маємо відомості про те, яку частину навчального матеріалу користувач пройшов (значення від 0 до 1). В результаті кожної показаної рекомендації ми можемо зіставити деяке число: -1 відповідає відмови від рекомендацій, 1 - переходу за посиланням без проходження курсу, 2 - переходу за посиланням і повного проходженню курсу, значення від 1 до 2 - неповного проходженню курсу, а 0 - відсутності реакції.

3.4.3 Формування видачі

Отже, ми отримали списки навчальні матеріали від декількох обробників, в яких кожний навчальний матеріал має вагу від 0 до 1, чим більше, тим більш вдало обробник оцінює рекомендацію. Постає питання, як їх найкраще комбінувати.

Можна було б просто для кожного навчального матеріалу скласти або перемножити ваги обробників, які його порадили. Але це означало б, що ми

вважаємо різні обробники однаково корисними, що, взагалі кажучи, може бути зовсім не так. Хотілося б оцінити кожен обробник коефіцієнтом корисності, використовуючи при цьому реакції користувачів на рекомендації. Таким вимогам задовольняє лінійна регресія.

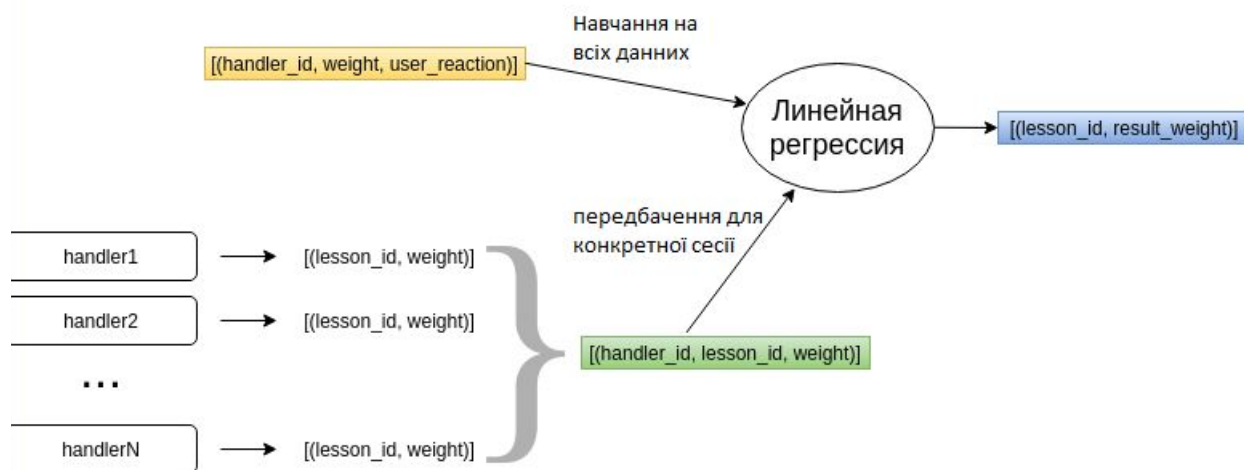


Рисунок 3.3 – Використання лінійної регресії

На рис. 3.3 схематично зображено цей процес. Регресійна модель навчається з використанням накопичених даних про те, як рекомендації різних хендлерів були оцінені користувачем. Виходячи з цього модель формує вектор ваг для обробників, вага обробника відповідає його корисності і впливає на його внесок у видачу. Далі, коли потрібна конкретна рекомендація, регресійна модель для кожного курсу комбінує ваги запропонували його обробників з використанням свого вектора ваг для них, і видає результуючий вага для кожного рекомендованого уроку. Нижче представлено формальний опис регресійній моделі.

3.4.4 Лінійна регресія

Нехай у нас є матриця $X = \{x_i^j\}_{i \in \{1..n\} j \in \{1..d\}}$, $x_i \in R^d$, рядки якої $x_i = (x_i^1 \dots x_i^d)$ називаються спостереженнями, а стовпці факторами, і стовпець значень цільової або залежною змінною $y = \{y_i\}_{i=1}^n$. Регресійна модель $y = f(X, b) + \epsilon$, де b - параметри моделі, а ϵ - випадкова помилка, називається лінійною, якщо залежність цільової змінної від факторів є лінійною, тобто $f(x, b) = b_0 + \sum_{i=1}^d b_i \cdot x_i$. Зазвичай для зручності b_0 (константу) вносять під знак суми, додавши в матрицю спостережень стовпець з одиниць: $x_i^0 = 1$ для всіх $i \in \{1, \dots, n\}$.

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1^0 & x_1^1 & \dots & x_1^d \\ x_2^0 & x_2^1 & \dots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^1 & \dots & x_n^d \end{pmatrix} \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \beta = \begin{pmatrix} b_0 \\ \vdots \\ b_d \end{pmatrix} \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

Використовуючи ці позначення, лінійну регресійну модель можна виписати як $y = X\beta + \epsilon$. Рішенням цього завдання будемо вважати стовпець $\hat{\beta}$, який мінімізує суму квадратів відхилень передбачаються значень від реальних, тобто $\hat{\beta} = \arg \min_{\beta \in R^{d+1}} \sum_{i=1}^n (y_i - x_i \beta)^2$. Такий спосіб апроксимації називається методом найменших квадратів (ordinary least squares) і є найбільш широко застосовується в контексті вирішення задачі лінійної регресії.

У нашому випадку ми можемо розглядати в якості спостережень, рядки матриці X , рекомендовані уроки будуть в якості ваги факторів цих спостережень,

які Хендлер призначили навчальному контенту, а в якості значень цільової змінної - оцінку рекомендації користувачем.

Таким чином, для контенту i_1, \dots, i_n , що був запропонований користувачам j_1, \dots, j_n відповідно, і для хендлерів 1, ..., 9 матриця спостережень X і стовпець значень цільової змінної (реакції користувача) y будуть виглядати наступним чином:

$$X = \begin{pmatrix} weight_{i_1 j_1}^1 & weight_{i_1 j_1}^2 & \dots & weight_{i_1 j_1}^9 \\ \vdots & \vdots & \ddots & \vdots \\ weight_{i_n j_n}^1 & weight_{i_n j_n}^2 & \dots & weight_{i_n j_n}^9 \end{pmatrix} \quad y = \begin{pmatrix} r_{j_1} \\ \vdots \\ r_{j_n} \end{pmatrix}$$

Вектор $\hat{\beta}$, який дає мінімальну помилку передбачення реакції користувача, містить індивідуальну вагу для кожного з хендлерів, на який потрібно помножити його рекомендацію. Якщо вага якогось із хендлерів відносно велика, значить, цей хендлер вносить позитивний внесок в рекомендації, і навпаки.

Для роботи рекомендаційної системи регресійна модель буде на регулярній основі навчатися на даних про реакцію користувачів на рекомендації. Ми будемо використовувати бібліотеку SciPy [10] для вирішення цього завдання і знаходження стовпчика $\hat{\beta}$. У цій бібліотеці вже реалізовано рішення задачі лінійної регресії.

Регуляризація після первинної реалізації рекомендаційної системи з лінійною регресією, яка знаходила стовпець ваг для хендлерів $\hat{\beta}$, що мінімізує квадрат відхилень реальної реакції користувача від запропонованої $\hat{\beta} = \arg \min_{\beta \in R^{d+1}} \sum_{i=1}^n (y_i - x_i \beta)^2$, стало помітно, що поступово вага одного

хендлера необмежено зростає, в той час як інші зменшуються, що в підсумку призвело до формування видачі практично тільки з результатів цього хендлера. Такий ефект називається позитивним зворотним зв'язком, і характеризується тим, що відхилення в результатах роботи системи впливає на її подальшу роботу, причому чим далі, тим більше результати зсуваються в бік цього відхилення.

Крім цього, так як Хендлер можуть пропонувати дуже схожий рекомендації, в наших даних також може бути присутнім проблема мультиколінеарності факторів, що тягне за собою слабку відособленість матриці X і, як наслідок, нестабільність рішення. В результаті ми отримуємо рішення, яке дає маленьку помилку на даних, на яких воно навчається, і велику на реальних даних. Ця ситуація називається перенавчанням (overfitting) моделі.

В якості вирішення проблеми перенавчання можна розглянути регуляризацію. Згідно книзі The Elements of Statistical Learning [8], основні способи регуляризації - це ласо (LASSO, least absolute shrinkage and selection operator [24]) і гребнева регресія (Регуляризація Тихонова, ridge regression [25]). Обидва ці методи змінюють вираз, яке ми мінімізуємо в процесі пошуку рішення регресії, додаючи до нього штраф на норму вектора.

У разі ласо-регуляризації використовується l_1 -норма і рішення знаходиться як

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^{d+1}} \left(\sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda \|\beta\|_1 \right).$$

У разі ж гребньовій регресії використовується l_2 -норма і рішення виглядає як

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^{d+1}} \left(\sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda \|\beta\|_2 \right).$$

В обох випадках параметр λ підбирається в процесі оптимізації. Використання регуляризації методом ласо зменшує всі β_i , а ті, що і так були відносно невеликими, стають дорівнюють нулю. Таким чином, метод ласо добре підходить для вибору значущих факторів (feature selection).

Метод гребньовій регресії також зменшує ваги факторів, але при цьому ніколи не зводить їх до нуля, якщо тільки не $\lambda = \infty$. Згідно роботі [17] ласо-регуляризація працює краще гребньовій в ситуації, коли кількість чинників значно перевершує число навчальних спостережень. У зворотному ж ситуації більш доречна гребнева регресія. Відповідно, для нашого випадку краще підходить саме вона. Її реалізація також присутній в бібліотеці SciPy.

3.5 Аналіз результатів

Для аналізу роботи вийшла рекомендаційної системи доцільно вибрати інтуїтивно зрозумілі метрики. Розглянемо основні класи метрик, що застосовуються для оцінки рекомендаційних систем[21]:

- Перевага користувачів: яку з безлічі систем користувачі використовують найбільш охоче. Використовується у випадку, коли є можливість провести порівняння, запропонувавши різні системи декількох груп користувачів. Метрики цього класу розраховуються для всіх зацікавлених систем, після чого порівнюються.

- Точність передбачення: застосовується в разі, якщо система проорокує відношення користувача до об'єктів (наприклад, виставляються ними рейтинги) або ймовірність використання користувачем контенту. До цього класу також відносяться метрики, що оцінюють якість ранжирування списку рекомендацій.

- Покриття: яку частину з цікавих користувачеві об'єктів системі вдалося передбачити. Зазвичай ускладнюється відсутністю повної інформації про інтереси користувача.

- Надійність: чи можна в цілому довіряти рекомендаціям. Дозволяє відокремити систему з в середньому непоганими рекомендаціями від такої, в якій на пару відмінних рекомендацій припадатиме десяток абсолютно не доречних.

- Інші метрики, такі як довіра користувача системі, новизна і різноманітність пропонованих рекомендацій та інші.

Також можливі різні режими оцінки рекомендацій:

- офлайн-оцінка: попередньо збирається інформація про користувачів, емулюється робота системи, результати оцінюються виходячи з тих же даних;

- штучні дослідження: спеціально відібраний групі користувачів пропонується опитувальник, який їм потрібно заповнити в процесі використання системи, також досліджуються накопичені в процесі дані про роботу системи;

- онлайн-оцінка: рекомендаційна система випробовується в «Бойових умовах».

У цій роботі будуть використовуватися оцінки, пораховані на основі тестових даних про використання користувачами системи в тестовому режимі. Досліджено дані на основі простих рекомендацій користувачеві на основі

інформації про нього, про рекомендації запропоновані по проходженню навчального матеріалу (контекстні рекомендації). У перших двох випадках рекомендації показуються користувачеві списком уроків, які можуть його зацікавити. У разі простих рекомендацій в цьому списку двадцять уроків, в разі контекстних - п'ять.

3.5.1 Прості рекомендації

Рекомендації для користувача показуються списком з 20 навчальних матеріалів. У користувача є можливість дізнатися, чому йому було запропоновано саме цей урок (причини беруться з обробчиків, що запропонували його), а також відмовитися від рекомендації конкретного навчального матеріалу, натиснувши на відповідну кнопку біля цієї рекомендації.

Для оцінки будуть використовуватися наступні метрики:

- Відсоток сесій, в яких було здійснено хоча б один перехід за посиланням або відмова від рекомендації. На жаль, так як рекомендації показуються на головній сторінці, ду мало хто їх дійсно бачать, тому цей відсоток буде невеликий. Подальші метрики будуть розглядатися тільки для сесій, на які була отримана якась реакція, позитивна (перехід по засланні) або негативна (відмова від рекомендації);
- Середнє число переходів по посиланню в сесії;
- Яку частину уроку в середньому проходять користувачі після переходу за посиланням;
- Скільки скоєно відмов від рекомендацій.

Були досліджені дані про роботу системи за тиждень. Результати представлені в таблиці 3.1.

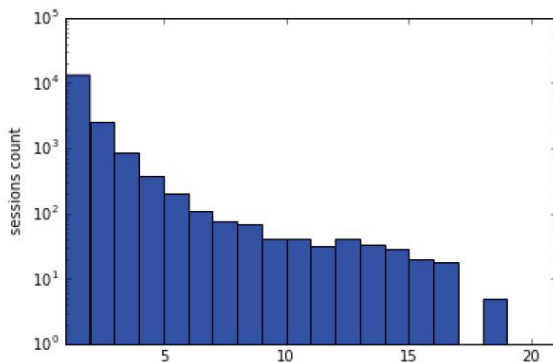
Таблиця 3.1: Результати для простих рекомендацій

Метрика	Значення
Число сесій	60
Число сесій з реакцією	41
Відсоток сесій з реакцією	34%
Число відкритих рекомендацій з 5	1.71
Пройдена частина одиниці навчального контенту	0.6
Число відмов від рекомендації	0

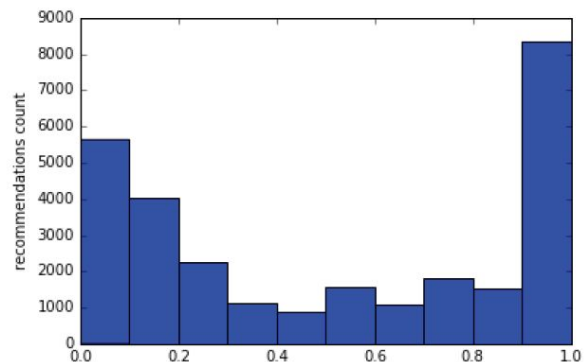
На рис. 3.4 представлені розподілу числа відкритих рекомендацій і пройденої частини уроку.

3.5.2 Контекстні рекомендації

У разі, якщо навчальний матеріал, пройдений користувачем, не включений ні в який курс, тобто немає явного наступного за ним навчального матеріалу, користувачеві пропонуються рекомендації, створені на основі інформації про нього. В даному випадку користувач напевно бачив рекомендації, що прямо відбиваються на відсотку рекомендаційних сесій з реакцією.



(a) Відкрито посилань



(б) долі рішень

Рисунок 3.4 – Розподіл для простих рекомендацій

У таблиці 3 також відображені результати за тиждень.

Таблиця 3: Результати для контекстних рекомендацій метрика значення

Метрика	Значення
Число сесій	35
Число сесій з реакцією	25
Відсоток сесій з реакцією	74.4%
Число відкритих рекомендацій з 5	1.48
Пройдена частина одиниці навчального контенту	0.62
Число відмов від рекомендації	0

Висновки до розділу

В даному розділі було розглянуто питання вибору мови програмування на якій буде розроблена рекомендаційна система, було визначено, які дані збирає система, також були поставлені вимоги до системи, обраний спосіб

агрегації даних у вигляді графів, розглянута чиста архітектура, описаний алгоритм роботи системи та як формуються дані для видачі.

Додатково було розглянуто питання вибору бази даних, яка буде використовуватися для збереження користувацьких даних.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
<p>Змістом даної роботи є створення веб застосунку, який облегчить процес навчання для студентів, бо кожен студент має різні вподобання, й постійно зайнятий пошуком інформації у напрямку, що їх цікавить. Для розвитку йому потрібно закріплювати існуючі та здобувати нові знання, вивчаючи нові технології та методи розробки.</p> <p>Університети не можуть дати всі необхідні знання, по всіх предметних областях й для розвитку своєї кар'єри студенту потрібно отримувати нові навички, які в університеті не викладають, тому студенти витрачають багато часу на пошук цих знань.</p>	<p>У навчальних закладах</p>	<p>Користувачами нашої системи будуть студенти, вони матимуть змогу переглядати список навчальних матеріалів, проходити їх та отримувати навички, що будуть відображатися у їхньому профілі. Під час проходження курсу, студентам може бути не зрозумілий деякий матеріал, й вони будуть користуватися пошуковими системами, для знаходження відповідей на свої запитання.</p>

Таблиця 4.2. Опис ідеї стартап-проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Дипломний проект	LinkedIn	Plural-Sight			
1	Швидкість роботи	Повільна	Швидка	Середня	Повільна обробка великої кількості даних	Простота архітектури	Накопичення інформації в одному місці
2	Зручність використання	Зручно	Зручно	Відсутня	Відсутність анімації	Відсутність реклами	Розроблений зовнішній вигляд для роботи через веб-сторінку
3	Вимоги до системи	Низькі	Високі	Високі	Швидкість роботи залежить від потужностей системи	Оптимізована робота через веб-додаток	Актуальність програми для старих систем
4	Кросплатформність	Наявна	Наявна	Наявна	Не весь контент оптимізований під мобільні пристрої	Налаштована система для роботи через головні веб-інструменти	Використання популярних фреймворків дозволила створити адаптивний дизайн

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Використання методу фільтрації вмісту	У рекомендаційних системах	Наявний базовий алгоритм	Повністю відкрита для роботи
2	Використання методу колаборативної фільтрації вмісту	У рекомендаційних системах	Наявний базовий алгоритм	Повністю відкрита для роботи
3	Використання методу гібридної фільтрації, що включає в себе фільтрацію вмісту та колаборативну фільтрацію	У рекомендаційних системах	Наявний базовий алгоритм	Повністю відкрита для роботи
<i>Обрана технологія реалізації ідеї проекту: 3</i>				

Висновок: технологічна реалізація продукту – можлива, вибрана метод №3, який може нам допомогти розробити якісний продукт з використанням комбінації технологій, тобто гібридна система з більш активним використанням фільтрації контенту і менш активною - колаборативною фільтрацією.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн./ум.од	600

Закінчення таблиці 4.4

3	Динаміка ринку	Стагнація національної економіки, хоча інвестують у розвиток ІТ. Темпи розвитку світової економіки позитивні
4	Наявність обмежень для входу	Відсутні. Конкуренти займають свої певні сфери, які висвітлюють функції, які вказані в дисертації стартапу, тільки як додаткові.
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні.
6	Середня норма рентабельності в галузі або по ринку, %	60

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим, держава інвестує в розвиток науки, що надає змогу реалізувати продукцію на національному ринку, оскільки на ньому немає конкурентів.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Студенти має різні вподобання. Для розвитку йому потрібно закріплювати існуючі та здобувати нові знання, вивчаючи нові технології та методи розробки.	Студенти	Стартап буде допомагати студентам у пошуку навчальних матеріалів, рекомендувати та дозволяючи доповнювати їх	.

Закінчення таблиці 4.5

2	Університети не можуть дати всі необхідні знання, по всіх предметних областях й для розвитку своєї кар'єри студенту потрібно отримувати нові навички, які в університеті не викладають.	Викладачі	Стартап буде рекомендувати викладачам додати матеріал, який користується попитом у студентів й також викласти доповнення до матеріалу лекцій	Зручність у використанні. Швидка робота системи. Спроможність швидко освоїти як користуватися системою.
---	---	-----------	--	---

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів
3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника.
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями що відсутні у компаній конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників.
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи кешування даних.
4	Грошова винагорода за рекламу	При достатньому попиту на систему кешування даних можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвитку продукту та оплати заробітної плати працівникам	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проект задля його подальшого розвитку.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим замінником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери вживання що не покривають інші товари-замінники; Кореляція цін у відповідності до товарів замінників; Різні типи ліцензій.
2	Рівень конкурентної боротьби: світовий	Всі продукти замінники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з клієнто-необхідною функціональністю; Налагодження маркетингу на основних Інтернет ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту.
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки ІТ додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки.
4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки.
5	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках.	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові і аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замін ники
	LinkedIn, PluralSight	Udemy	Official Website, Google Play, Apple Store	Офіційний веб сайт	Курси компанії Udemy
Виснов ки	Прямі конкуренти намагаються сконцентрувати ся на інших ринках та напрямках.	Потенційні конкуренти мають дуже специфічну клієнтуру до якої зазвичай входять корпоративні клієнти. Конкуренція може відбутися тільки продажах на корпоративний рівень клієнтів	Постачальники диктують умови збереження даних, які захищають приватність користувачів. Також постачальники не дають змогу зловживати етичними нормами.	Клієнти можуть диктувати умови на ринку тільки через повідомлення на форумах або в полі відгуків в точках продажу додатку	Розповсюдження навчальних матеріалів для різного рівня підготовки

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Прагматичність	Через запуск стартапу система буде не дуже складно з точки зору архітектури перший час. Але необхідно залучити викладачів для створення контенту й студентів для наповнення рекомендаційної системи інформацією
2	Зручність	Оскільки стартап розробляється на багатьох платформах з різною шириною екранів, то зручність використання системи на різних пристроях буде відігравати не малу роль у спроможності конкурувати з іншими гравцями ринку.
3	Швидкість роботи	Швидкість роботи відіграє велику роль для користувачів, оскільки вони не будуть готові чекати декілька хвилин на виведення результату роботи додатку.

4	Оптимізація	Якщо додаток буде дуже часто видавати помилки при роботі, то користувачі не будуть вважати додаток надійним.
5	Налаштування під користувача	Різні люди мають різні звички, які вони використовують, наприклад, якщо є люди, які люблять працювати за додатком де є темні кольори, а є такі люди, які люблять світлі кольори. Можливість редагувати зовнішній вигляд додатку надає значну перевагу серед конкурентів.
6	Відкритість вихідного коду	При наявності вихідного коду будь-який продукт має перспективи розвиватися у багатьох напрямках, особливо таких, які можуть бути неочевидні на перший погляд.
7	Приватність	Інформація про оцінки студентів мають бути приватними, й викладачі матимуть доступ лише до загальних даних
8	Технічна підтримка	Якщо технічна підтримка компанії буде працювати своєчасно та швидко, то це допоможе зберегти репутацію компанії на відміну від конкурентів, де їй не приділяють увагу.

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін системи кешування мало змінних даних

№	Фактор конкурен тоспромо жності	Бал и 1- 20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Прагмати чність	15	Plural Sight	Udemy	Linke dIn				
2	Зручність	8			Plural Sight	Linked In	Udemy		
3	Швидкіст ь роботи	5					Udemy	LinkedI n	Plura l Sight
4	Оптиміза ція	5					Plural Sight	Udemy	Link edIn
5	Налашту вання під користув ача	10				Plural Sight	LinkedI n	Udemy	

6	Відкритіст ь вихідного коду	20	Plural Sight	LinkedIn	Udemy				
7	Приватніст ь	20	Udemy	Plural Sight	LinkedI n				
8	Технічна підтримка	15			LinkedI n	Udemy	Plural Sight		
9	Документа ція	15	Plural Sight	Udemy	LinkedI n				
10	Ціна	15		Udemy	Plural Sight	LinkedI n			

Таблиця 4.12. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – Прагматичність системи через її легкість роботи; – Простота у використанні; – Наявність відкритого вихідного коду; <p>Збереження приватності інформації користувача.</p>	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Неоптимізованість алгоритму; – Швидкість роботи системи; – Зручність.
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Зворотній зв'язок з клієнтурою компанії для спроможності розвивати проект в інші напрямки. 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – Невірні показники рекомендацій, при недобросовісних оцінках студентів

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс - наявний	2-3 місяці
2	Реклама	Залучення власних коштів для реклами товару	1-2 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1-3 місяці

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Студенти	Присятнуя	Високий	Майже відсутня	Легка
2	Викладачі	Відсутня	Середній	Відсутня	Середня
3	Школярі	Присутня	Низький	Відсутня	Легка
Які цільові групи обрано: 1, 2					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є студенти та викладачі. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для масової аудиторії в цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії).

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
--------------------------------------	---------------------------	--	---------------------------

Закінчення таблиці 4.15

Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів	Проведення реклами, освітлення унікальної функціональності через інтернет ресурси та інші канали, контакт напряду з споживачами; формування лояльності і прихильності споживачів	Зниження ступеню замінності товару; Прихильність клієнтів; Відмітні властивості сервісу; Відмітні характеристики сервісу;	Стратегія диференціації
---	--	---	-------------------------

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	Компанія частково копіює характеристики товару конкурента, основна ціль компанії розробка нового унікального функціоналу, з підтримкою основного функціоналу конкурентів	Стратегія заняття конкурентної ніші

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Зручність	Диференціація	Спроможність економити час на вивченні навчального матеріалу	Спроможність легко вивчати контент
2	Відкритість вихідного коду	Диференціація	Перспектива розвитку проекту	Розвиток в науці
3	Документація	Заняття конкурентної ніші	Саморозвиток	Рекомендації навчальних матеріалів

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Повторення пройденого матеріалу	Перелік навчальних матеріалів	Матеріал вносять викладачі

2	Отримання нових знань	Рекомендація навчальних матеріалів	Викладачі додають навчальний матеріал, що користуються попитом
---	-----------------------	------------------------------------	--

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Система кешування малозмінних даних на мові програмування Java		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Зручність	Нм	Е
	Швидкість роботи	Нм	Тх
	Оптимізація	Нм	Тх
	Ціна	Нм	Е
	Документація	Нм	Тл
	Технічна підтримка	Нм	Тх
	Приватність	Нм	Тх
	Налаштування під користувача	Нм	Ор
3. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент			

В/Нв – відчутні/невідчутні; М/Нм – монотонні/немонотонні; Пр/Нпр – параметричні/непараметричні; Вр/Тх/Тл/Е/Ор – вартісні/ технічні/ технологічні/ ергономічні/ органолептичні; О/К/С – обов'язкові/ кількісні/ сюрпризні

Таблиця 4.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
150– 300грн./од.	600 – 800 грн./од.	4 000 – 14 000 грн./міс.	100 – 800 грн./од.

Таблиця 4.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Студенти будуть купувати продукт групами. Психологи будуть купувати товар на корпоративному рівні. Туристи будуть купувати товар поодиночці.	Можливість скачувати додаток влюбий час, у будь-якому місці.	2 рівня (посередник + клієнт)	Роздріб

Таблиця 4.22. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Студентство	Форуми, навчальні заклади	Навчання	Показати можливість користування продукцією	Рекламне звернення спрямовано до потенційних клієнтів, де показуються плюси користування системою
2	Викладачі		Викладання	Показати перспект користування систе	

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі

товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки по розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

ВИСНОВКИ

Ефективна особиста система рекомендації навчальних матеріалів не тільки зменшує інформаційне перевантаження, рекомендуючи відповідний навчальний матеріал для своїх студентів, а також надає їм необхідну інформацію в необхідний час та запропонує необхідний шлях.

В магістерській дисертації було зроблено аналіз існуючих методів рекомендації, розглянуто використання рекомендаційних методів в існуючих системах та представлено оцінку рекомендацій. Було розглянуто підхід для розробки рекомендаційної системи, метою якої є підтримка студента через систему управління навчанням, для знаходження відповідного матеріалу для покращення знань студента, також був обраний тип персоналізованої навчальної системи.

Додатково в дисертації було визначено, що обраний метод рекомендації навчальних матеріалів має вдалі характеристики в рекомендації, що допомагають студентам вибрати відповідні навчальні матеріали. Проте якість рекомендацій також залежить від добросовісної оцінки студентами навчальних матеріалів.

Загалом якщо говорити про рекомендаційні системи навчальних матеріалів, то вони мають високі перспективи, бо кожен користувач, в нашому випадку студент, має індивідуальні потреби та конкретні вимоги. Деякі студенти сильно мотивовані і люблять вчити нове, в той час як інші студенти віддають перевагу специфічним гайдам та структурованому

шляху. З щоденним ростом комп'ютерних та інтернет технологій, багато закордонних університетів використовують системи керування навчанням для підтримки викладання та навчання. Ці системи поширюють різні навчальні матеріали, такі як статті, курси, відеоматеріали тощо. У деяких випадках, ці системи можуть бути покращені впровадженням підтримки персоналізованих рекомендацій.

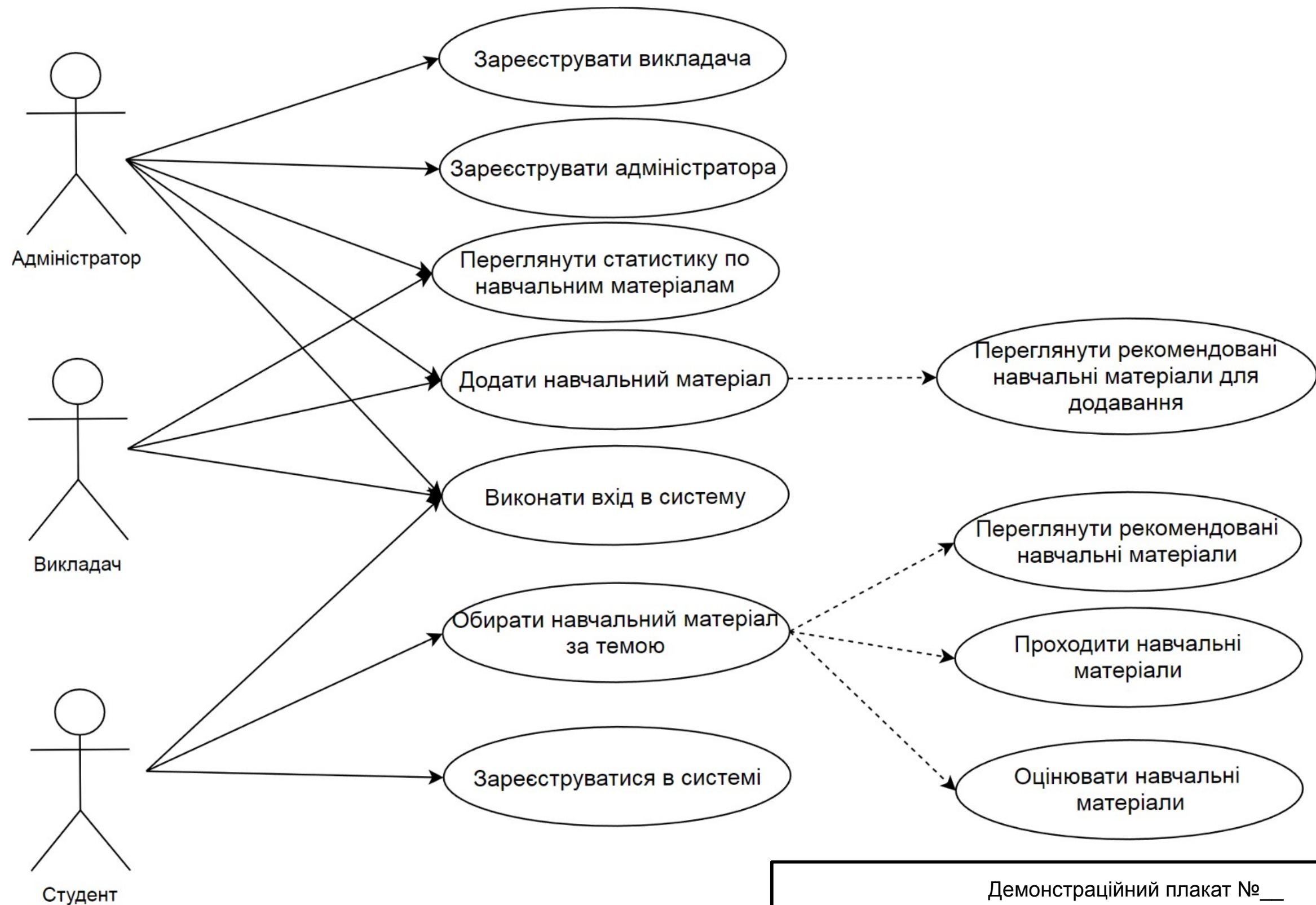
ПЕРЕЛІК ПОСИЛАНЬ

1. Персоналізована рекомендаційна система в LinkedIn
URL:<https://engineering.linkedin.com/blog/2016/12/personalized-recommendation-s-in-linkedin-learning> (дата звернення 10.11.2018)
2. Персоналізуйте ваш досвід навчання, досягайте бізнес цілей простіше
URL:<https://www.pluralsight.com/product/channels> (дата звернення 8.9.2018)
3. Adomavicius, G. and Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6), pp.734-749.
4. Burke, R., 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4), pp.331-370.
5. Candillier, L., Jack, K., Fessant, F. and Meyer, F., 2009. State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access Techniques for Improved User Modeling*.
6. Kramár, T. and Bieliková, M., 2012, September. Dynamically selecting an appropriate context type for personalisation. In *Proceedings of the sixth ACM conference on Recommender systems* (pp. 321-324). ACM
7. Santos, O.C. and Boticario, J.G., 2011. Requirements for semantic educational recommender systems in formal e-learning scenarios. *Algorithms*, 4(2), pp.131-154.
8. Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A. and Schmidt-Thieme, L., 2010. Recommender system for predicting student performance. *Procedia Computer Science*, 1(2), pp.2811-2819.

9. Ricci, F., Rokach, L. and Shapira, B., 2011. Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). springer US.
10. Jones Eric, Oliphant Travis, Peterson Pearu et al. SciPy: Open source scientific tools for Python. — 2001—. — [Online; accessed 2015-12-06].
URL: <http://www.scipy.org/>.
11. Pazzani, M.J. and Billsus, D., 2007. Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer, Berlin, Heidelberg.
12. Poorni, G., Student, P.G., Balaji, K. and DeepthiNivetha, C., A Personalized E-Learning Recommender System Using the Concept of Fuzzy Tree Matching.
13. Щербак Д.В., Сирота О.П. Система рекомендації навчальних матеріалів. Вчені записки Таврійського національного університету імені В. І. Вернадського. Том 29(68) №6 2018
14. JavaScript. URL: <https://developer.mozilla.org>
15. Spring Framework. URL: <https://spring.io>

ДОДАТКИ

Діаграма використання

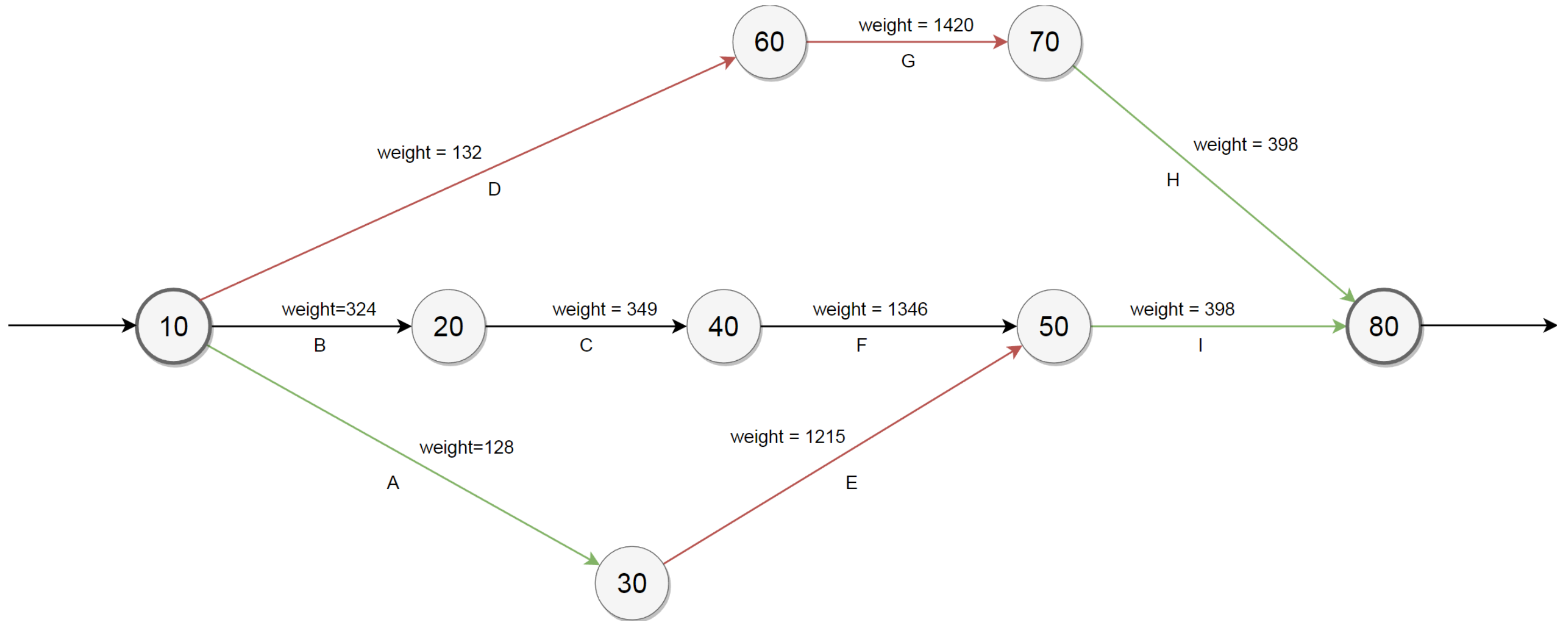


Демонстраційний плакат №__
до магістерської дисертації на тему
„Система рекомендацій навчальних матеріалів для персоналізації
навчальних матеріалів”

Розробив: _____

Прийняв: _____

Граф переходів

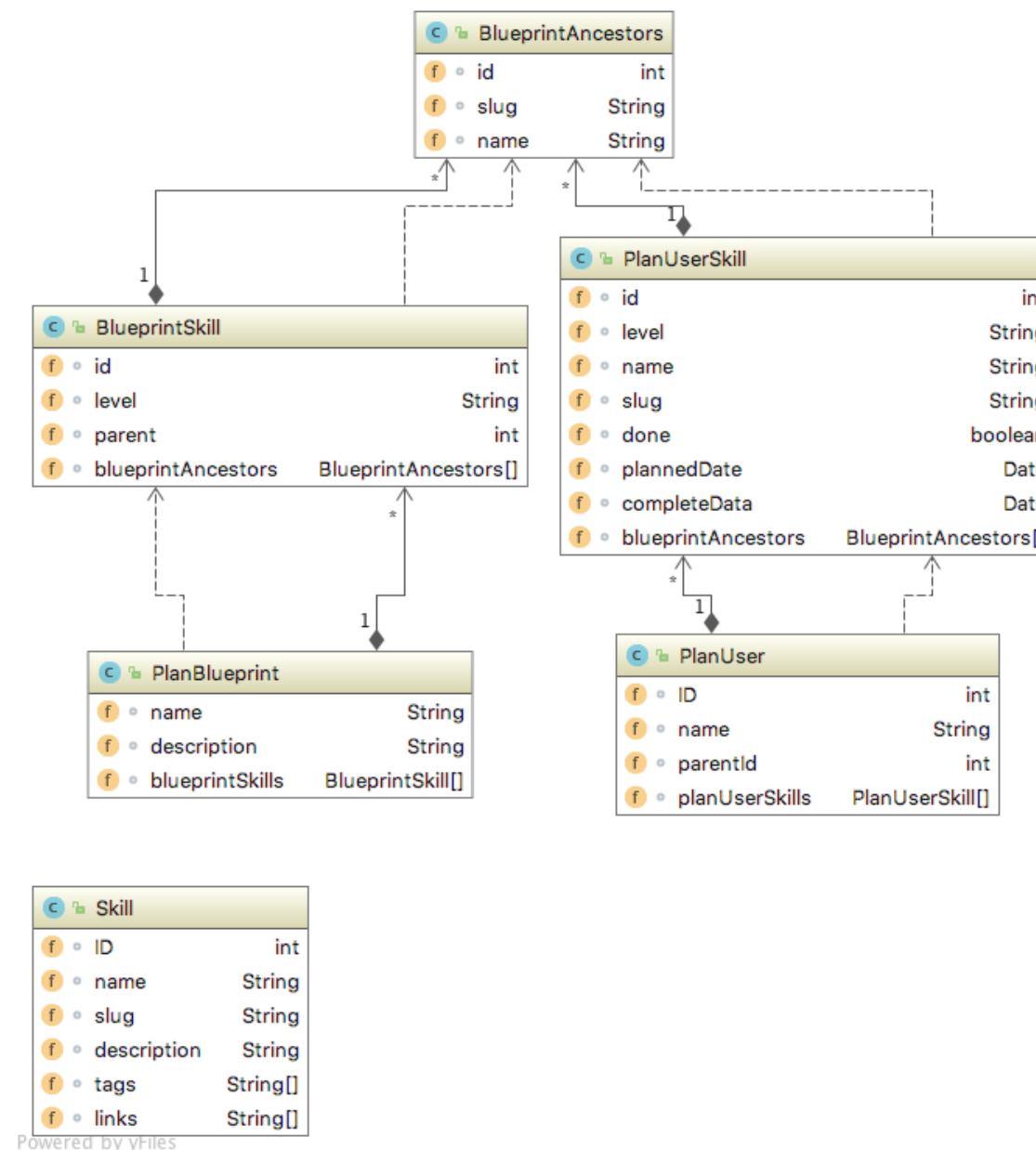


Демонстраційний плакат №__
до магістерської дисертації на тему
„Система рекомендацій навчальних матеріалів для персоналізації
навчальних матеріалів”

Розробив: _____

Прийняв: _____

Схема БД навчальних матеріалів

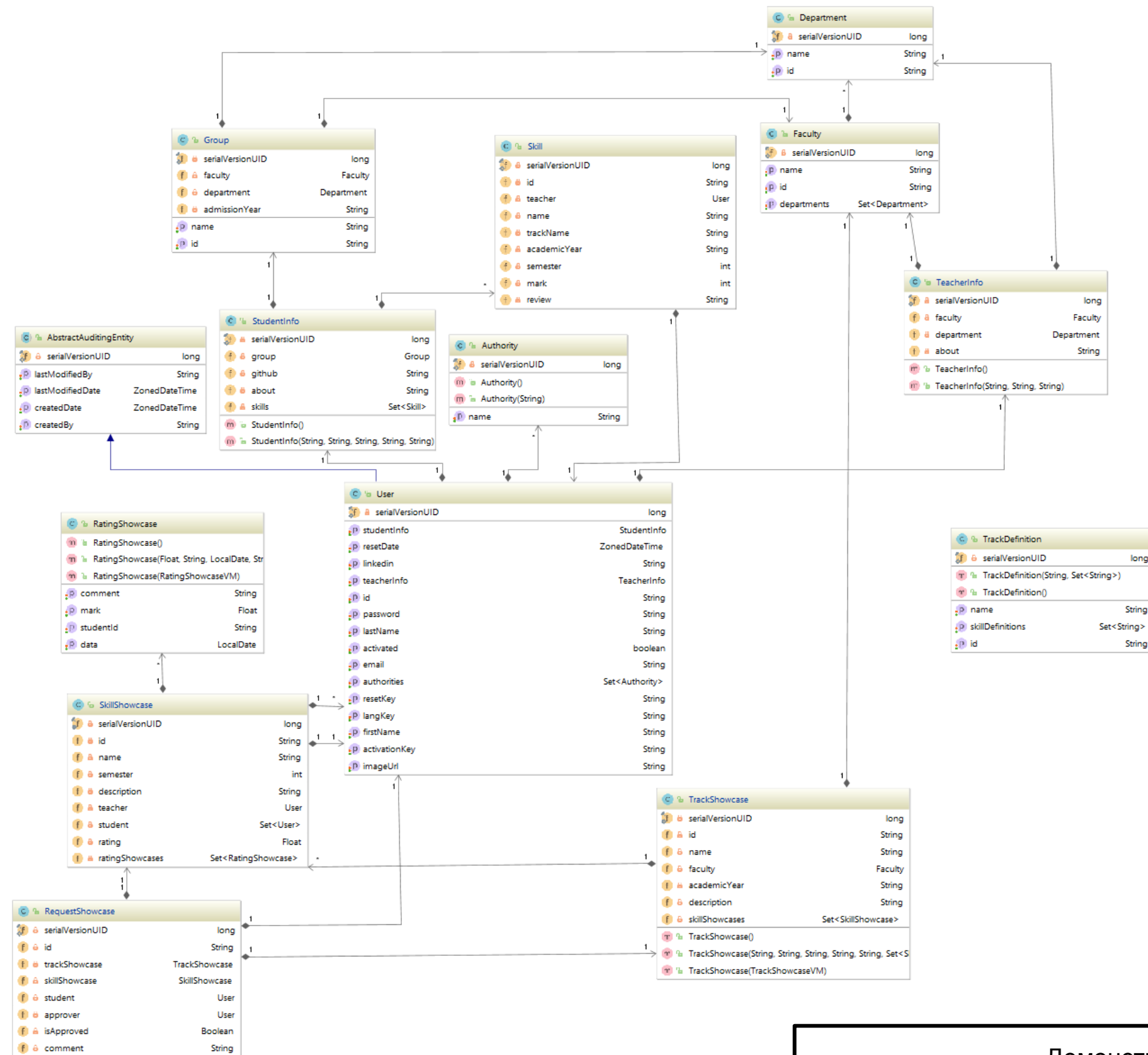


Демонстраційний плакат №__
до магістерської дисертації на тему
„Система рекомендацій навчальних матеріалів для персоналізації
навчальних матеріалів”

Розробив: _____

Прийняв: _____

Схема класів навчального закладу




Демонстраційний плакат №__
до магістерської дисертації на тему
„Система рекомендацій навчальних матеріалів для персоналізації
навчальних матеріалів”

Розробив: _____

Прийняв: _____

Інтерфейс програми

[← Back](#) [Add Skill](#)



Зінський Микола

Programming skills

JS

74.67

JAVA

62.50

C#

78.33

Student's skills

JS (2014-2017)

Node js Core

(2014-2015), 1 semester

74

Node js Core

(2014-2015), 1 semester

74

JQuery

(2016-2017), 2 semester

76

JAVA (2015-2017)

Core

(2015-2016), 1 semester

65

Spring

(2016-2017), 2 semester

60

C# (2012-2015)

.Net Core

(2012-2013), 2 semester

95

ASP.NET

(2014-2015), 1 semester

70

ADO.NET

[← Back](#)

Course C#

Faculty - FIOT

Academic Year - 2012-2013

Teachers -

Overall rating: 4.5/10

4.50

Course rating

.Net Core: 5

5.00

WPF: 4

4.00

Description:

C# course includes wpf, asp.net

.Net Core WPF

.Net Core

Console app development

Users						
Create a new user						
Email	Language	Profiles	Created date	Modified by	Modified date	
system@localhost	Activated	en	13/03/17 18:37	23/05/17 22:27		View Edit Delete
anonymous@localhost	Activated	en	13/03/17 18:37	23/05/17 22:27		View Edit Delete
admin@localhost	Activated	en	13/03/17 18:37	23/05/17 22:27		View Edit Delete
user@localhost	Activated	en	13/03/17 18:37	23/05/17 22:27		View Edit Delete
some@email.com	Activated	en	19/03/17 16:04	admin@localhost	04/04/17 00:00	View Edit Delete

Демонстраційний плакат №__
до магістерської дисертації на тему
„Система рекомендацій навчальних матеріалів для персоналізації
навчальних матеріалів”

Розробив: _____

Прийняв: _____